Mapping Strategic Thought and Action in Developing Disruptive Software Technology: Advanced Case Study Research on How the Firm Crafts Shared Vision

Hugh M. Pattinson
University of Technology, Sydney, Sydney
P.O. Box 123, Broadway NSW
Australia
Hugh.Pattinson@uts.edu.au
(Corresponding Author)

Arch G. Woodside
Boston College
Boston College, Carroll School of Management, Department of Marketing,
140 Commonwealth Avenue,
Chestnut Hill, MA 02467
USA
(arch.woodside@bc.edu).

Keywords: Mapping, Strategic, Thought, Software, Development, Case-Study, Hermeneutic, Research, Development, Commercialization, Innovation

June 2007

Abstract

This case study uses two storytelling methods for analysis – an advanced hermeneutic framework and an extended form of decision systems analysis (DSA) incorporating cognitive mapping – to explore the strategic thought in building a software house around a philosophy of "best practice" application development. The account explores decision-making for turning an ambitious vision (automation of enterprise-based sales and marketing activities) into a focused application (pricing configuration software), and then into a large software house (Trilogy) offering enterprise e-commerce suites. Trilogy's management team's decision-making was heavily influenced by a strong perception that the organization needed to take risks to achieve "critical mass" in anticipation of a convergence of "back-office" and "front-office" applications into one market. The advanced case study also addresses Trilogy's transition of its applications into Internet environments, plus the transformation of the organization from a product-orientation to a strictly industry-based business and application development perspective. Trilogy has not compromised its strict application development philosophies, but has incorporated its fast cycle time (FCT) methods into its industry-focused business divisions – and now actually offers the method as a set of services.

Introduction

The article presents and applies a hermeneutical framework (Arnold & Fischer 1994; Thompson, Pollio, & Locander, 1994; Thompson 1997) in research on B2B decision making. The case study research in the present report follows Woodside, Pattinson, and Miller's (2005) hermeneutic template and Huff's (1990; also see Huff & Huff 2000) mapping strategic thought templates. This article describes conceptual and research tools for achieving deep sensemaking of what happened and why it happened—including how participants interpret outcomes of what happened and the **dynamics** of emic (i.e., transformations in informants' own interpretations of what happened and why it happened) and etic (i.e., transformations in researchers' views about what happened and why) sensemaking. This report includes doing fresh "re-research"—that is going back to original informants to learn their further reflections on what happened and why

events happened, along with how these informants interpret earlier researchers' reports that followed from earlier case research studies on the informants' thinking and behavior (cf. Langley, Mintzberg, Pitcher, Posada, & Saint-Macary 1995),

Dynamic sensemaking relates to and advances from hermeneutical research. This article defines **hermeneutic research** as the inclusion of multiple rounds of informant-researcher interpretations of the dynamics of a specific situation framing-problem-decision-action-outcome by reflective analysis of autonomous text and multiple interviews of the same and different persons in different time periods. Harvard Business School (HBS) cases describing the histories of enterprises along with specific problems-actions-outcomes for these firms and are examples of autonomous text that may be incorporated into a hermeneutic framework. The present article includes re-interviewing informants participating in interviews for the original HBS case report and reporting these informants' interpretations of the original researchers' case report; the present article includes collecting additional data relating to the decision process and outcomes examined in the original case as well relevant data from subsequent time periods. Thus, the present report includes informants' interpretations of researchers' interpretation of prior informants' decisions and actions. Prior reports of multiple rounds of interviewing informants that include informants interpreting researchers' findings do appear in the B2B literature (e.g., Howard and Morgenroth 1968; Howard, Hulbert, & Farley 1975), Woodside and Samuel 1981). The inclusion of different sets of researchers, the systematic collection of additional data not included in the original report, and the retrospective focus represent a unique contribution by the present article.

Background on the Firm

Trilogy Inc., of Austin Texas provides enterprise software focused on reducing the costs associated with "front-end" or "selling chain activities", including sales and marketing. Trilogy emerged in 1991 from a group of Stanford University students pursuing the development of software to configure and price computer hardware parts.

Trilogy was selected for research on mapping strategic thought because the firm provides rich detail on software application development, both in the initial HBS case study accounts, and from the respondents within the company in follow-up personal interviews conducted for the present report. Strictly speaking, Trilogy's software application is not classified by Christensen and Raynor (2003) as disruptive technology. However, Trilogy was a pioneer in developing applications to specifically reduce sales and marketing costs that evolved into B2B E-commerce systems that may in 2007, arguably be now classified as disruptive technology – or at the very least powerful sustaining technology.

The Trilogy case is particularly worth reading by executives desiring to understand more about how highly trained and developed professional staff can make a big difference in product development. The Trilogy story provides further valuable lessons including developing expertise in the following areas:

- Rapidly transforming application development toward Internet/Web-based development
- Changing a company from a horizontal to a vertical industry marketing focus.

The original HBS case study was written by Robert D. Austin (see Trilogy 2003a, 2003b, Trilogy Corporation 2003) who has established expertise in research and executive curriculum development in information technology management, and more generally on the management of knowledge-intensive activities. He has written on these subjects in more than four books

The case study for the project traces Trilogy from its inception from an idea to address a specific marketing problem through to competing with large ERP organizations in the late 1990s and is extended to cover a new phase of industry-focused application development and delivery in 2001-02. The initial decision systems analysis (DSA, see Capon & Hulbert, 1975) model, events chronology map, and the three cognitive maps for this case were developed from extracts drawn from Austin's (1998) HBS study "Trilogy (A)". Austin, with Mandel, wrote an additional short note for the Case study, "Trilogy (B)" addressing the spin-off of several Internet-based on "Dot.com" businesses from Trilogy in 1999-2000 (Mandel & Austin 2000).

The main purpose of Austin's (1998) and Mandel & Austin's (2000) case study was to explore risk-taking in terms of creating and rapidly growing a new business and instilling such a

culture in the business, and particularly in software development. Austin outlined the creation and rapid development of Trilogy from 1991 through to 1998, while Mandel & Austin note a list of spin-off businesses out of Trilogy in 1999-2000. The case is relevant to this project because Austin provides a substantial account of application development and its central role in Trilogy's corporate culture. Trilogy's early applications were developed mainly in C++ but were transitioned across to an Internet-based platform using Java and XML from 1997-99.

Austin's (1998) account of the application is sufficient for the development of a representative DSA model and to undertake further analysis on application development and delivery. The DSA model and maps for this case were updated, plus extension questions on dominant logic, shared vision, key leverage points and strategic marketing issues were addressed through interviews. The interview participants were one of the key players in the HBS case study, who is still with Trilogy, and the current Vice President of Development at Trilogy (see Franke 2002a, 2002b; Hyams 2002).

Trilogy Case Study Contributions and Limitations

The Trilogy case study explores the decision-making associated with creating an organizational culture of "best-practice" application development processes. The case offers insights into combining a philosophy of "best-practice" with risk-taking in order to grow a company rapidly to prepare for an eventual and imposing competitive showdown with larger and more established corporations.

The case study also provides insights into decision-making from the original conception of the product through to the creation and rapid growth of Trilogy and on to a transition into a provider of vertically-focused (industry-focused) applications. There are substantial insights into the decision-making associated with the high standards set for product-oriented application development, and how those high standards are translated into an industry-focused application development environment. Trilogy's strict and high-quality application development standards make it the most internally-focused software house in this project. Decision-making processes associated with such an environment provide an interesting comparison with other software houses in this project.

Interviews for the Trilogy case study captured substantial insights beyond Austin's (1998) and Mandel & Austin's (2000) case study, particularly on Trilogy's new industry-focused application development cycles. Additional material provided by Trilogy enabled the most comprehensive insights into the actual application development cycle out of all the cases for this project. Creation and validation of a set of DSA models plus cognitive mapping outputs represents an effective mapping of, and contribution to strategic sensemaking. In the Trilogy case study this approach is applied to turning an idea into an application, in conjunction with turning a small start-up business into a substantial and competitive enterprise software provider.

Trilogy had a small period of time in which to grow fast enough to be ready to face severe competition from much larger and more established ERP software houses that could stake out its niche and incorporate it into their broad enterprise-wide systems.

A selection of the cognitive maps provides insights into both Trilogy and its competitors' perceptions of an emergent convergence of niches into larger enterprise-wide application suites. DSA models and selected maps for the Trilogy case provide unusual insights into a very strong logic for setting innovative application development processes and cycles. Trilogy has quite structured and unusual requirements regarding the skills expected of developers, and has a special system of recruitment and training to instill shared vision and empowered decision-making at the individual developer level in projects.

The mapping of decision-making associated with transitioning from a "best-practice" product-driven business and application development approach to a vertical industry-focused approach, is a an extended contribution from additional insights gained from the interviews stages at Trilogy. The Trilogy advanced case study contributes to practice providing valuable insights into establishing "best-practice" application development groups. The establishment of a strong shared vision around programming and testing excellence through recruitment and training can be drawn from the Trilogy case.

The Trilogy case study provides some practical insights into transforming a software house from a strong product development focus across to industry-focused business groups and application development. The Trilogy case study benefited from a complete set of multiple interviews with multiple respondents who were directly involved in the original case study on the company, to validate the original set of DSA models, events chronology maps and cognitive maps. The new DSA models and new cognitive maps developed from interview material were validated within the timeframe allowed for this project. There may have been additional benefit in finding more description about the transition to straight Internet-based applications during 1997-99. However, this issue was partly addressed through the collection of additional information on Internet development and Internet business initiatives, during the interview process.

Trilogy Case Study Hermeneutic Analysis

Five levels of hermeneutic analysis were developed to address decision-making issues, events, and linkages in the Trilogy case study. Exhibits 1 and 2 present the first and second levels of analysis, while Exhibit 3 is the final level of analysis.

Exhibit 1 Level I Analysis of Trilogy Case

Emic 1 view:

- Liem and t identified problem s with hardware com paniesm anagingtheir selling and delivery processes
- Sales and Marketingcosts typically represented over 40% total com puter hardware vendor expenses – and were open to substantial autom ation
- Developm ent of pricing configurator application for hardware com panies
- Expanded to be a com plete "front-end" enterprise application
- To be the best application developm ent com pany in the world
- 15 or 20 successful software com panies could be built around the talent in Trilogy's developm ent organization
- New philosophies on application developm ent based on sm all team s of "superprogram m ers"
- Deep concern about reaching "critical m ass" to com pete against large "back-office" ERP providers

↓

1991-1998 Initial mental model:

- Focus on Autom ation of Sales & MarketingActivities
- Aim to cut hardware vendor Sales & MarketingCosts by 2 % per year
- Excellence in Application Developm ent through using sm all team s of "super program m ers" m anagingwhole projectswith frequent version builds and autom ated testing
- Trilogy University established to further develop superprogram m ers
- Rapid expansion of application to be a com plete front-end e-com m erce application – before the back-end ERP vendors took that space from trilogy

Decisions-actions:

- Developed configurator application (1991-1992); Sold to Selected Com puter H ardware V endor plus other m anufacturingcom panies (1992-1994): expanded application to support all elem ents of Sales & Marketing (1995-1998); Faces Head-On Competition from m ain ERP V endors (1997-1999)

Exhibit 2 Level II Analysis of Trilogy Case

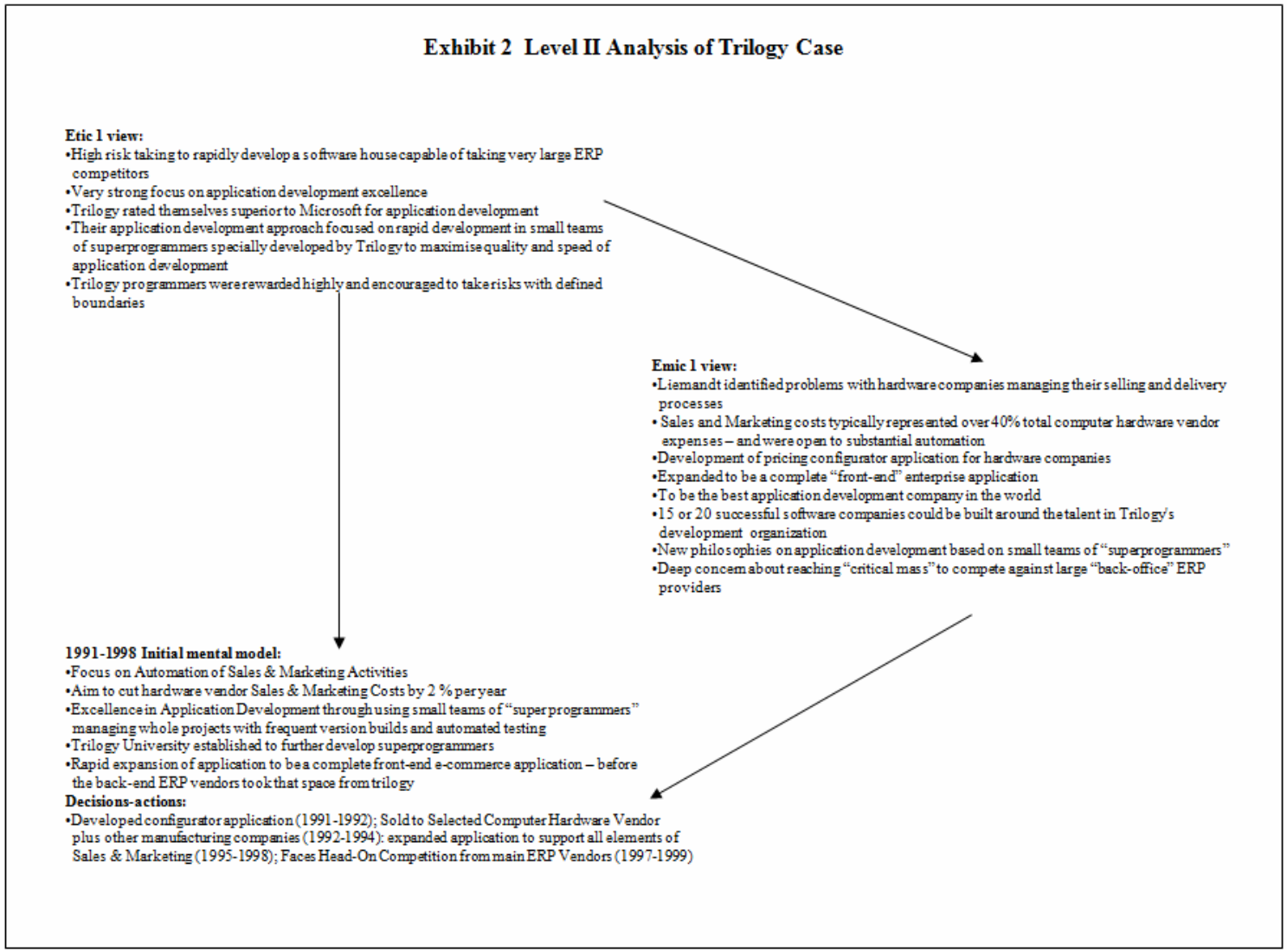
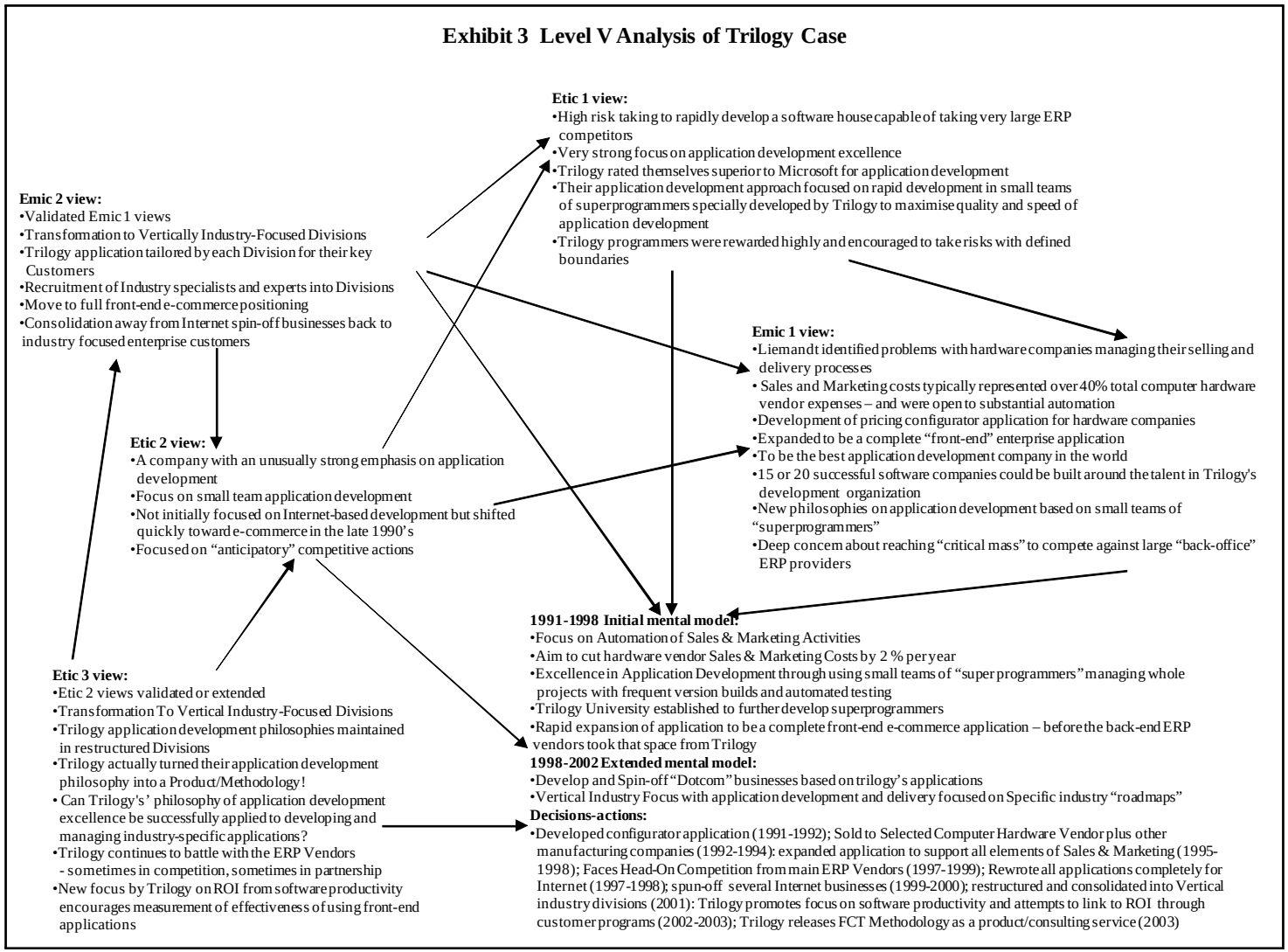


Exhibit 3 Level V Analysis of Trilogy Case



The original case study by Austin (1998) serves as the etic 1 report of the case study – although there are elements of emic 1 (i.e., reports by informants in the original case study report) reporting in the form of direct quotes from decision-makers in the account. Quotes and perspectives from decision-makers were encapsulated the Level 1 emic analysis in Exhibit 1. Key case study issues as articulated by Austin (1998) were added to the emic 1 data in the Level II analysis (see Exhibit 2) and presented as etic 1 (reports by researchers in the original case study report) data. These issues include the following points:

- High risk-taking to rapidly develop a software house capable of taking very large ERP competitors
- Very strong focus on application development excellence
- Trilogy rated itself as superior to Microsoft for application development
- Its application development approach focused on rapid development in small teams of superprogrammers specially developed by Trilogy to maximize quality and speed of application development
- Trilogy programmers were rewarded highly and encouraged to take risks with defined boundaries.

The Level II analysis also explored mental models and decisions/actions associated with these issues.

The third level of analysis is an etic 2 (new reporting by new round of researchers) representation based on the current researcher's summarization of event milestones and the emic 1 sensemaking views identified in the data in the Austin (1998) case study. Austin's (1998) account focuses on the decision-making associated with identifying the automation and reduction of sales and marketing costs as a major unmet enterprise need and developing a comprehensive e-commerce application suite over time to address this need. The account also highlights application development excellence of a core value in Trilogy's corporate culture. The account was sufficient for the development of an initial (etic 2) representative DSA model and to undertake further analysis on application development and delivery.

The initial DSA, event chronology, and cognitive maps reported below offer details supporting the third level analysis. Etic 2 perspectives are mainly based from the initial DSA model, initial events chronology map, and initial cognitive maps.

Key etic 2 issues include the following points:

- A company with an unusually strong emphasis on application development
- Focus on small team application development
- Not initially focused on Internet-based development but shifted quickly toward e-commerce in the late 1990s
- Focused on "anticipatory" competitive actions.

The etic 2 DSA model and maps for the case were updated following extensive questioning of the accuracy and completeness of the original Austin's (1994) case – the collected data represents emic 2 material and was added during the fourth level of analysis. Thus, additional (emic 2) data were collected for etic 3 description and interpretation of the decision-making process as reported in the original case study, and for a period of four years beyond that reported by Austin (1998) and three years beyond Mandel & Austin (2000). Emic 2 data consists of responses from interviews with the then principal developer of Trilogy's "SalesBuilder" application, and the current (as early 2003) VP of Development at Trilogy (see Franke 2002a & 2002b; Hyams 2002a & 2002b).

The fifth and final level of analysis (see full analysis in Exhibit 3) with the addition of etic 3 data summarizing emic 2 interpretations of mental models and events covered in the original case study as well as the work completed for the etic 2 interpretation – including the DSA, event chronology, and cognitive maps developed for the etic 2 interpretation. The revised DSA, event chronology, and cognitive maps presented in subsequent sections below follow from the emic 2 interpretations, and these maps are part of the etic 3 interpretation.

The new data from the emic 2 and etic 3 rounds of interpretation validated etic 2 data and highlighted new insights related to the following topics:

- Transformation to vertical industry-focused divisions
- Trilogy application development philosophies maintained in restructured divisions
- Trilogy actually turned its application development philosophy into a Product/Methodology
- Can Trilogy's philosophy of application development excellence be successfully applied to developing and managing industry-specific applications?
- Trilogy continues to battle with the ERP vendors - sometimes in competition, sometimes in partnership
- New focus by Trilogy on ROI from software productivity encourages measurement of the effectiveness of using front-end applications

Overall, the five-level hermeneutic framework for the Trilogy case study provided a strong and comprehensive framework for capturing and extending dynamic sensemaking for Trilogy, for both the development of a pricing configurator application into a comprehensive front-end e-commerce application suite, in tandem with the rapid development of a small business into a robust medium-sized enterprise.

DSA Model

The first DSA model was developed from a review of the whole Austin (1998) case study. The supporting note by Mandel & Austin (2000) was dropped from the analysis as it only addressed a list of new business ventures and not application development and delivery. The section in Austin's (1998) case study under "Developing the Product" was the main source for developing a DSA model (see Austin 1998, 7-8). The account discusses development philosophies, issues and approaches that could be regarded as distinctive to Trilogy. Figure 1 presents the original DSA model and is representative of Trilogy application development during 1996-99.

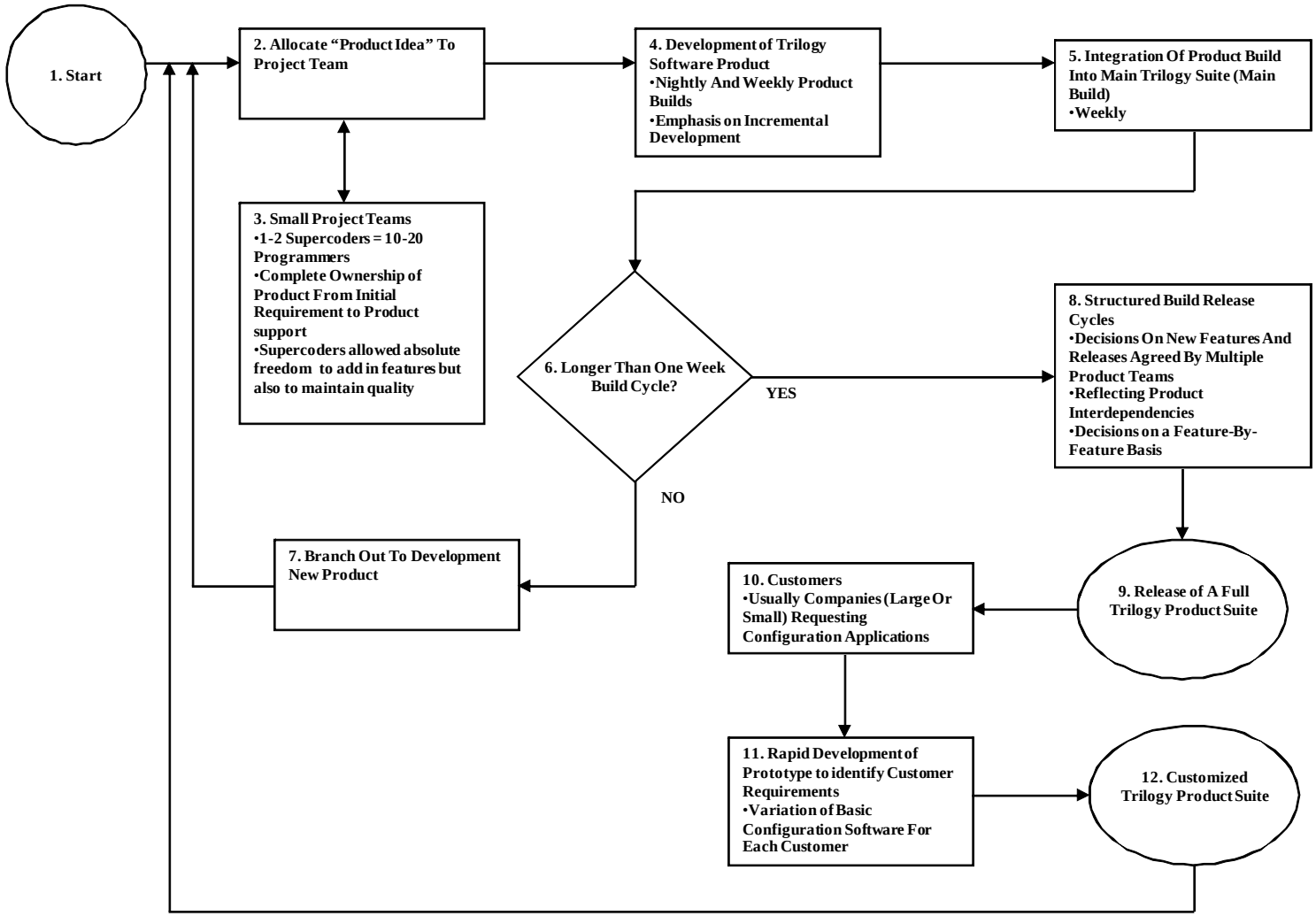


Figure 1 Summary DSA Model For Trilogy Application Development (1998)
Created from data in Austin (1998)

The application development commenced with the allocation of a "Product Idea" to a project team (box 2). A distinctive feature of Trilogy's application development cycle is the concept of very small projects composed of 1-2 "Supercoders" (box 3). Trilogy's management believed that one to two highly experienced and motivated programmers could do the work of 10-20 standard programmers. These Supercoders were given complete ownership of a project right through the whole application development cycle. Trilogy also believed in high levels of remuneration, bonuses and compensation, and ongoing training for the Supercoders, who were also accorded superstar status within the organization.

Components of the application were built or released as new development versions either on a nightly or weekly frequency. Updates to the development version were usually incremental (box 4). Components were integrated into the application weekly (box 5). A significant decision-point each week for the project team was whether a component or application would be likely to take longer than a weekly build cycle (box 6). If "NO" then the project team(s) can go onto new product development (box 7).

If "YES" then the project teams developed structured build release cycles incorporating decisions relating to new features and new releases (box 8). A set of collective decisions were made among a number of project teams working on application components or full applications, understanding that there were significant interdependencies between their projects. It is likely that this collective decision-making approach was actually incorporated into the application development cycle right from the commencement of the cycle. A full Trilogy Product Suite was released (box 9). However, that was not the end of the development cycle. Customers often required major customization of Trilogy's applications into their own configuration systems (box 10).

A significant number of these customer requests added up to an extension of the application development cycle, whereby Trilogy rapidly prototyped a variation of the basic configuration software tailored for a customer request (box 11). The customized application was in effect a new version of the application and could be regarded as a completion point in the application development cycle (box 12). Trilogy's application development cycle was supported by a rapid decision-making system with a mix of very small groups (one to two coders) working in concert, with high collective empowerment, with a vision of an integrated application suite. There was provision for rapid customization beyond the basic application, which is worthwhile being viewed as part of the overall application development cycle. Interviews with Franke (2002) and Hyams (2002) confirmed the DSA model for Trilogy without any modifications for Trilogy's application development cycles up to 1999.

New DSA Model (2001-02)

The interviewees for the Trilogy case requested that a new DSA model be created for Trilogy's application development cycles from 2001-02. The new DSA model reflects Trilogy's commitment to vertical (industry) focused business management and product development. The interviewees provided substantial insights in the development of the new DSA model. Detailed product development cycle information was drawn from company documents on Trilogy Product Management Methodology, and is used but not directly referenced, with permission from the interviewees at Trilogy. These inputs were compiled into a vignette which is presented in Exhibit 4 with the new summary DSA model presented in Figure 2.

Exhibit 4 Trilogy's Industry-Focused Application Development

TRILOGY GOES VERTICAL – INDUSTRY-FOCUSED ENTERPRISE SOLUTIONS

In 2000, Trilogy restructured from a horizontally focused application vendor toward using a Software Development Methodology designed to quickly conceptualise, create and deliver highly verticalized applications. Previously, from 1993-1999, Trilogy's application development strategies were based on Trilogy Product Roadmaps.

The Executive Team established an Operations Group with five vertical divisions focused on key enterprise customers and products:
* Automotive
* Computer
* Telecommunications
* Financial Services
* New Business

Trilogy has a two-dimensional matrix of management and delivery resources, located in Core company-wide Groups and within each Division with deep industry experience. These resources cover:
* General Managers plus Functional Vice-Presidents
* Development Staff (Division-Specific)
* Consulting
* Human Resources
* Finance
* Solutions Marketing and Business Development
* Product Management and Presales Resources

Trilogy's Fast Cycle Time (FCT) Software Development Methodology is now based on delivering vertical applications within Industry Maps set up for each division for a two-year timeframe. There are four phases in this methodology:

1. Product Ideation
During this phase, Trilogy's Division Managers with their Solutions Marketing and Product Marketing teams generate ideas for products, creates a vision for the product and the Industry, applies an investment justification process to the vision and product, and develops an Industry Roadmap supported by a Business Case. All product conceptualization and development is driven with a vertical industry focus.

Exhibit 4 Trilogy's Industry-Focused Application Development (Cont'd.)

2. Product Planning
During this phase, the capabilities of the product will be identified and sequenced for delivery. Product features will then be expanded into actual development schedules including actual release features and initial estimates at release dates. During this phase, Product Management takes over with Development Teams in driving product planning.

3. Product Development
This phase consists of gathering the detailed requirements for the product, and commencing development using Trilogy's Fast Cycle Time methodology. Although Product Management and Development Teams are driving the product development process, chartered customers associated with specific Trilogy divisions may be involved in testing early versions of the product.

4. Product Delivery
Product Delivery is defined as beyond Beta Release. There is extensive interaction between Product Management, Development Teams, charted customers, and solutions marketing groups. The final product will be delivered during this phase. Once in production, the product will be maintained with patches, bug fixes and sub-releases. Consultants may work with chartered customers in deployment of new products.

An extended version of steps in Trilogy's Software Development Cycle is outlined below:

1. Identify and Define Product Concepts (Product Ideation)
* Industry Analysis
* Analyze Against Current Trilogy Industry/Product Maps
* Driven by Solutions Marketing/ Product Management; some input from Division General Management

2. Evaluate Product Concepts (Product Ideation)
* Product Vision and Product Concept
* Validate Product Concepts Against Industry Maps
* Develop Business Case
* Driven by Solutions Marketing/ Product Management; Division General Management

3. Sequence Capabilities and Define Product Features (Product Planning)
* Define and Sequence Product Capabilities
* Product Features against Industry Maps
* Driven by Product Management with input from Development Teams

Exhibit 4 Trilogy's Industry-Focused Application Development (Cont'd.)

4. Create Release Definition (Product Planning)
* Determine Release Features and Scope
* Determine Resource and Time Estimates
* Define Release Review Cycles
* Agreed and Finalised Between Product Management; Development Teams; Division General Manager; Some input from Nominated "Charter Customers"

5. Define Release Requirements (Product Development)
* Identify Potential Product Requirements
* Prioritize Based On Business Value, & Decide on Release Contents
* Requirements Inputs (From Consultants
* Development Release Plan
* Undertaken by Product Management; Development Teams; some input from nominated "Chartered Customers"

6. Develop Product (Product Development)
* Iterative Development Via Feedback and Short Development Cycles
* Product Locked Once Released to Beta
* Iterative Updates
* Develop Launch Development Plan at Beta Stage
* Active Involvement from Product Management; Development Teams; Solutions Marketing (For Product Roll-out and Marketing); feedback from nominated "Chartered Customers" associated with testing

7. Release Product (Product Delivery)
* Product Release Certification Process
* Certification For Specific Platforms
* Marketing Launch
* Driven by Development Team; Solutions Marketing

8. Monitor Product (Product Delivery)
* Support including Bug fixes and Minor Enhancements
* Feedback From Installed Base
* Set Up For New Products And/Or Next Release
* Directed by Product Management with product fixes and sub-releases completed by Development Team

This vignette was prepared from Hyams, Chris (2002a & 2002b) for the present case study.

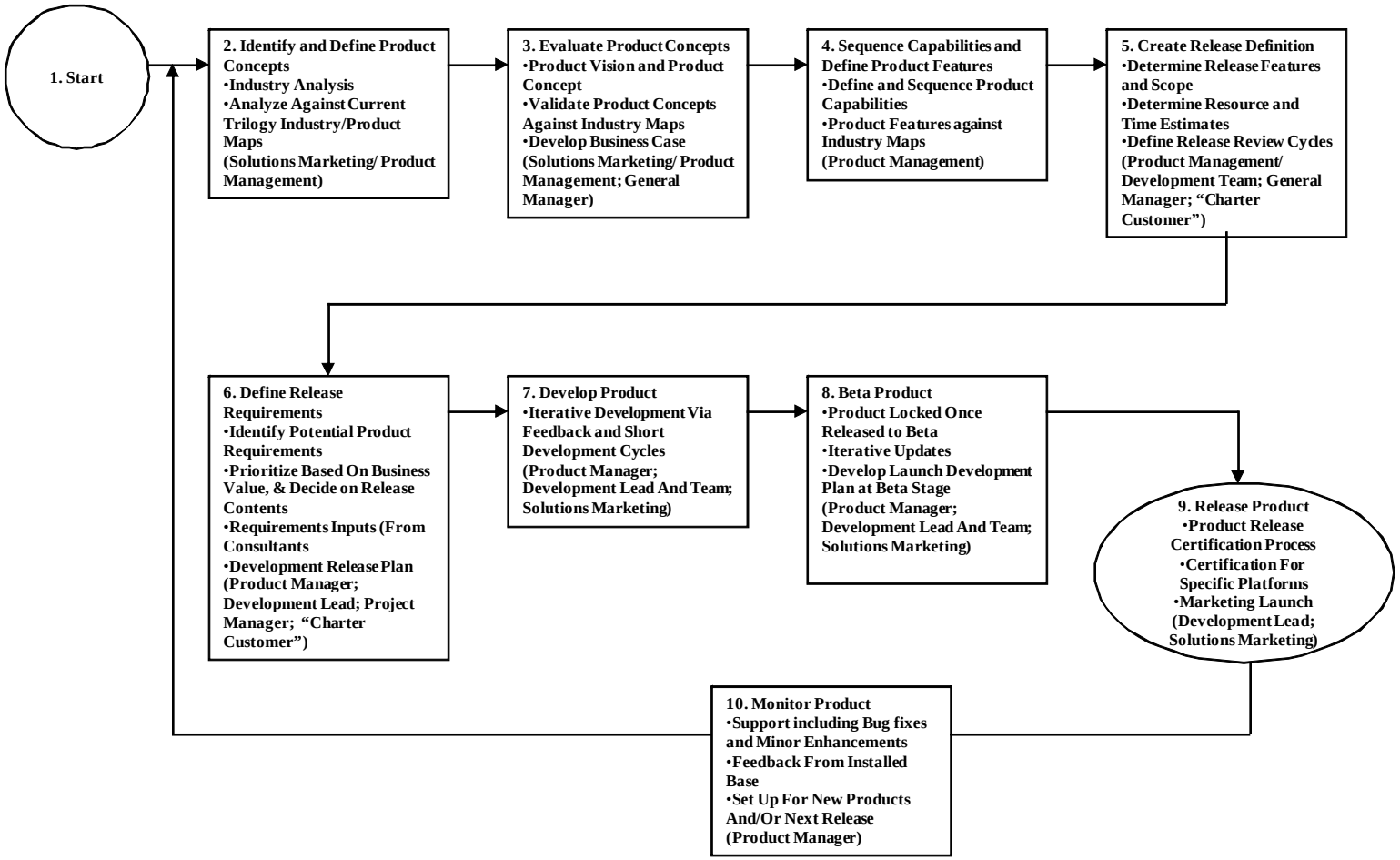


Figure 2 New Summary DSA Model For Trilogy Application Development (2001-2002)
Created from data in Austin (1998); Franke (2002); Hyams (2002)

Trilogy's industry-focused approach to application development is evident at the commencement of the application development cycle as all product concepts that are identified and defined are subjected to rigorous industry analysis (see box 2). Solutions Marketing and Product Management are responsible for the decision-making associated with mapping concepts against existing Trilogy industry and product maps.

Industry Group General Managers with Solutions Marketing and Product Management conduct further validation of concepts against industry maps and develop business cases for the concepts (box 3). Product concepts are transformed into actual product features and capabilities by Product Management (box 4). The defined product features are validated against industry maps, reinforcing that each of the Trilogy industry-based business divisions are directly involved in specific application development. Several groups are involved in creating a Release Definition, which includes release features and scope, resource and time estimates, and release review cycles (box 5). Product Management and the Project Development Team will create the actual documentation and content, but will consult closely with the relevant industry group General Manager for specific release dates. Selected "Chartered Customers" will also be consulted, particularly with regard to product features and scope.

Trilogy's application development cycle has a specific step for the definition of release requirements which has more detail and further prioritization of product features than the release definition (box 6). There is an additional assessment of product features based on business value, and inputs from Trilogy and other consultants. Outputs from these assessments are then finalized into a Development Release Plan. Product Managers working with the Project Team and "Chartered Customers" drive the finalization of the Development Release Plan.

The actual product development phase is iterative and through short development cycles similar to the weekly/monthly build approach in the original DSA model (box 7). Product Management works with the Project Team, but also includes Solutions Marketing in this phase. The direct involvement of industry-based solutions marketing persons in the actual product development phase is a distinctive feature of Trilogy application development cycle, when compared with other software houses in this project.

A distinct beta product milestone occurs where product features are locked in and a Launch Development Plan is developed (box 8). There are still iterative updates based on testing the product at this stage. Product Management, with the Project Team and Solutions Marketing, are all active in beta testing and creating the Launch Development Plan. Trilogy has formal processes for product release (box 9). Trilogy's products are formally certified for actual release and for specific platforms (specific operating systems and/or vendor's computer systems). A formal marketing launch occurred for the product release. Certification processes are driven by the development teams, while solutions marketing drive the marketing launch.

Trilogy has formal post-release processes for product monitoring, including bug fixes and minor enhancements (box 10). Customers provide feedback on product performance and recommended fixes. Product Managers oversee product monitoring and take inputs for setting up the next application development cycle. The new DSA model represents a significant departure from the original DSA model in that it recognizes that almost all application development is undertaken within the Business Divisions and is focused right from conception on the industry map. Business requirements are focused within that Division, rather than being a generic application that was customized at the end of the application development cycle.

There are still some application development inputs from Trilogy executives in charge of development, but that is rapidly contextualized into industry/division-specific decision-making processes. The new DSA model still reflects an internal focus for applications required for Trilogy's own application development. However, there is significant third-party interaction between Trilogy and selected "chartered customers" and consultants.

Events Chronology Map

Austin's (1998) "Trilogy (A)" case study provides a chronology of the genesis and rapid growth of Trilogy from 1990 through to 1998-99. An events chronology map was created incorporating key events during this period (see Figure 3). Additional symbols were employed

for the map including the flowchart symbol for a delay or problem and a six-pronged box for a solution to the delay or problem.

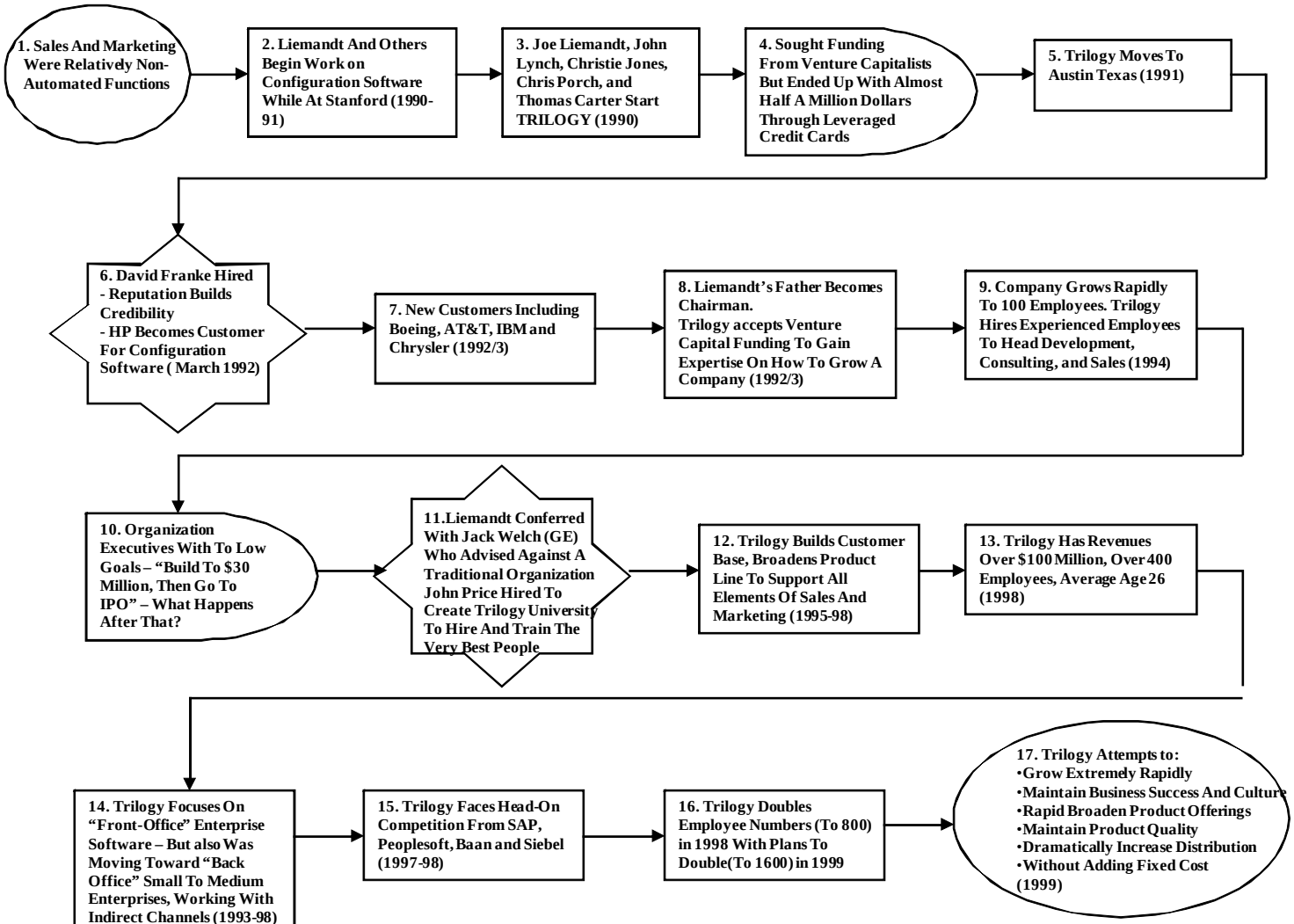


Figure 3 Events Chronology Map – Trilogy
Created from data in Austin (1998)

Joe Liemandt recognized that, compared to other functions with large computer hardware companies, sales and marketing were relatively non-automated functions (box 1). Liemandt believed that software to assist with the configuration of complex computer systems could greatly contribute to cutting costs in the sales and marketing functions.

While at Stanford University, Liemandt, with some other students, began work on configuration software in 1990-91 (box 2). Liemandt, Lynch, Jones, Porch and Carter set up Trilogy in 1990 (box 3). They sought funding from venture capitalists but were regarded as too young, and ended up leveraging their own credit cards up to almost half a million dollars. (box 4). During 1991, Liemandt decided to move the new company to Austin, Texas, to be closer to his ill father (box 5). The move enabled Liemandt to more easily hire David Franke, who was an experienced and well-known software developer (box 6). Franke's strong reputation in software development was a contributor to Trilogy signing up Hewlett-Packard (HP) as a customer in March 1992.

Trilogy rapidly built on its success with providing a configurator solution for HP, through signing up new customers including Boeing, AT&T, IBM and Chrysler in 1992-93 (box 7). Trilogy gained extra business development knowledge through the appointment of Joe Liemandt's father, Greg, as Trilogy's Chairman in 1992. Greg Liemandt had recently been a senior executive with GE and a Chairman of UCCELL. Trilogy turned down several venture capitalists' offers of funding, but chose other specific VCs to gain expertise on how to grow a company rapidly (box 8). By 1994, Trilogy had grown to 100 employees, having hired experienced development, consulting and sales managers (box 9). However, Trilogy's executives were struggling with establishing reasonable growth targets for the company – and when to go public (or to IPO), and where the company might go after an IPO (box 10). Through his father's links with GE, Joe Liemandt was able to confer with Jack Welch the well-known CEO of General Electric (GE) about these issues (box 11).

Welch advised that Trilogy not establish a traditional organization structure, but to focus on hiring the best people to develop the best products. John Price was hired to set up the Trilogy University that would hire and train high-quality developers. Over the three years 1995-98, Trilogy built a substantial customer base, while broadening its product line beyond configuration to cover all elements of sales and marketing (box 12). By 1998, Trilogy was generating revenues of over $100 million, with over 400 employees whose average age was just 26 (box 13).

Also, through the period 1993-98, Trilogy moved toward end-to-end solutions, working with indirect channels partners such as system integrators (box 14). Expansion from "front-office" into "Back-Office" activities put Trilogy on a collision course with ERP vendors including SAP, PeopleSoft, Baan and Siebel (box 15). During 1998, Trilogy's employee numbers doubled and were expected to double again in 1999 (box 16). Austin (1998) highlights several issues facing Trilogy's executive team for 1999, around continued rapid growth, broadening product offerings, maintaining product quality, and maintaining business success and a risk-taking development culture (box 17). Trilogy also needed to dramatically increase distribution of its products without adding fixed costs to its overall business model. It seemed that Trilogy had to stay on an exponential growth curve forever.

Updated Events Chronology Map

The original events chronology map was validated during the interview stage without modification up to 1999, and includes reference to Mandel & Austin's (2000) Trilogy (B) case study. The map timeframe was extended to 2002, and, Figure 4 presents an updated events chronology map.

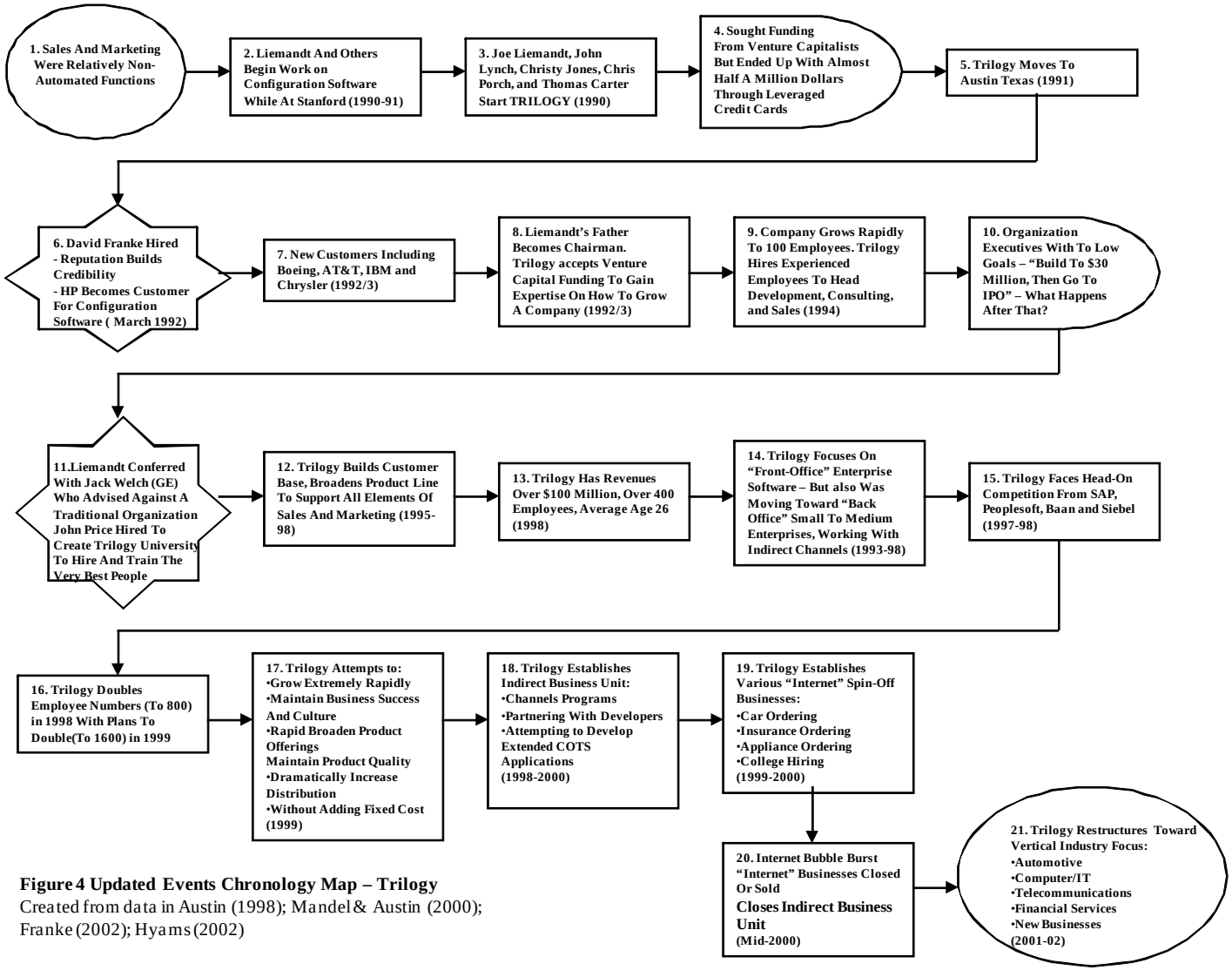


Figure 4 Updated Events Chronology Map – Trilogy
Created from data in Austin (1998); Mandel & Austin (2000); Franke (2002); Hyams (2002)

During 1998-99, Trilogy strove to extend its product range toward a full enterprise-wide commercial-off-the-shelf (COTS) e-commerce suite (box 18), through the establishment of an indirect business unit. Trilogy attempted to partner with developers to rapidly extend its application suite. Trilogy also attempted to tailor its packages for easier use by developers. Vantage was noted as a key partner in 1998-2000.

Trilogy established various Internet spin-off businesses in car ordering, insurance ordering, appliance ordering and college hiring from 1999-2000 (box 19). These spin-offs leveraged off Trilogy's configuration application knowledge, or developer training skills, or from enterprising Trilogy programmers and executives. who wished to branch out into new business ventures.

However, by mid-2000, with the Internet "bubble" bursting, Trilogy closed or sold most of its Internet businesses (box 20). The indirect business unit was also abandoned, mainly because of the complexity of attempting to establish an extended COTS application suite. This experience reinforced a view within the company that developing a general COTS application suite just like the ERP suppliers was definitely not a path that Trilogy should travel on. Trilogy's own growth was halted and the company undertook major restructuring and consolidation into 2001. Trilogy's response to a falling information technology market was to restructure toward a vertical/industry focus in 2001-02, creating five Business Divisions (box 21). Each Division manages its own application development, consulting, marketing and its own business functions. The five Business Divisions created include Automotive, Computer/IT, Telecommunications, Financial Services, and New Business. Several issues emerging from the original and updated events chronology maps are explored further though cognitive maps.

Cognitive Maps

Cognitive maps created for the Trilogy case study were designed to complement DSA models and events chronology maps through additional exploration of selected decision-making issues and contexts. Three sections were extracted from Austin's (1998) case study for further text analysis and creation of cognitive maps to complement the analysis presented in the original DSA model and original events chronology map.

The three initial cognitive maps created for the Trilogy case study addressed the following themes:

- Early development (1991-94)
- Industry and competition
- Software development.

During the interview process, requests were put forward for an additional cognitive map reflecting updated insights from 2000-01: Industry-focused application development (2000-01)

The first cognitive map is an extended insight into the creation and early growth of Trilogy. The second cognitive map provides insights into industry competition. Part of the second map parallels Trilogy's early growth, but about half of the map focuses on Trilogy's competition as it rapidly grew between 1993 and 1999. The third cognitive map amplifies logic and philosophy behind Trilogy's distinctive application development methodology for 1993-99, but is also relevant through to 2002. The additional cognitive map highlights insights into Trilogy's industry-focused application development in 2001-02.

Cognitive Map 1 – Early Development (1991-94)

The cognitive map was developed from the "From founding to market leadership" section in "Company Background" section, Austin (1998, pp. 2 – see Exhibit 5). Figure 5 presents this cognitive map. The map covers insights into the identification of an opportunity in 1990 to automate activities within sales and marketing functions, onto addressing the opportunity through the development of a configurator application, and the creation of a start-up company. The map also provides insights into rapid growth of Trilogy through to 1994.

Exhibit 5 Text Extract - Trilogy - Early Development (1991-94)

Liemandt had come to Stanford knowing he wanted to start a software company. As an undergraduate, he had spent many hours researching the industry and thinking about where the best opportunities might be. In and around school work, he did consulting to pay bills and to stay on top of the latest in information technology practices. Eventually, experiences from consulting clicked with the research.

It seemed to Liemandt that hardware vendors had difficulty delivering their products with the right equipment. His consulting clients frequently received computers with missing or incompatible components. Selling and delivery processes for these complex products appeared to be largely manual and fraught with error. This observation prompted Liemandt to analyze the income statements of computer product companies, comparing spending patterns with the degree of automation in each company function. What he found surprised him (see Exhibit 1).

Companies typically spent only 8 to 10 percent on General and Administrative costs, which had been extensively automated over the years. Research and Development, also highly automated in most high-tech companies, accounted for slightly more spending, about 10 to 15 percent. Manufacturing was a similar story: mature cost saving technologies had been applied to reduce spending (less cost of goods sold) to 20 to 25 percent of the firm's total expenses. What remained after deducting these major categories was the more than 40 percent of expenditures that were mostly related to Sales and Marketing-an area which, surprisingly, was not very automated. If automation of the "selling chain" allowed companies to put an additional 2 percent of revenues on the bottom line (a number which Trilogy now considers reasonable), that would be worth literally billions of dollars. It was a potentially huge market that had not yet been targeted by anyone.

While still in school, Liemandt and the others had begun working on configuration software, which incorporated complex if-then rules into a tool that would prevent mismatches between incompatible product parts. They continued this work after school, into 1991, always sure they were on the verge of solving the configuration problem that would finally give them a completed product. Companies like Hewlett-Packard (HP) and Digital were working on their own "configurators," which added urgency to Trilogy's efforts. Liemandt presented their ideas to venture capitalists, but none

3 Some of the factual material in this section was found in "Dream On," by Karen E. Starr, Selling Power, October, 1997, Vol. 17, No. 8; and "Holy Cow, No One's Done This!" by Josh McHugh, Forbes, June 3,1996.

would invest in a company composed entirely of barely 20-years-olds. To stay afloat, the team leveraged more than 20 credit cards, managing to borrow almost half a million dollars in cash advances. Liemandt describes the mood in the days before the company had revenues or product:

At the beginning, nothing worked. We lived failure every quarter. The product never worked. We'd sit around thinking "this is just ridiculous, why are we continuing?" We were living in a state of failure, but we had this hope, this shining star that wouldn't go out. What kept us together was the vision that this was a huge opportunity; we just needed to make it work. That, or the fact that we were yelling at each other the whole time.

In 1991 the company moved to Austin, Texas so that Liemandt could spend more time with his father, Gregory Liemandt, who had been diagnosed with a fatal illness. By this time Trilogy had an early product and had applied for patents covering its algorithms, but the company still had no customers. They were working hard to generate interest in their software, but nothing was working. HP, a key potential customer, sent a particularly discouraging letter saying, in essence, "we already have a configurator and don't need your product."

Exhibit 5 Text Extract - Trilogy - Early Development (1991-94) (Cont'd.)

Meanwhile, however, being in Austin enabled Trilogy to hire David Franke, a software developer with an industry-wide reputation, from a research consortium in Austin. With Franke on board, the company suddenly had new-found credibility. Silicon Graphics became the first customer, signing a small deal. Within months, HP was back, this time offering $3.5 million for software and support services. The deal was consummated in March of 1992. At the time, Trilogy had eight employees.

When HP signed, everything changed for Trilogy. Software that was good enough for HP was good enough for a lot of other big companies, also. The floodgates opened and Boeing, AT&T, and, eventually, IBM and Chrysler became customers (the IBM deal alone was worth $25 million). Also significant: Liemandt's father, a former GE executive and chairman of UCCELL, who had called his son a moron for squandering his Stanford education, agreed to become Trilogy's chairman, a position he retained until he passed away in 1993. Trilogy accepted funding from two venture capitalists, not because the company needed the money, but because it needed the expertise on how to grow a company that those firms could offer. Venture investors who had refused to fund Trilogy in the early days came calling-and were turned away. Liemandt retained more than 50 percent ownership.

As orders rolled in, Trilogy staffed up. The company grew rapidly to around 100 employees. They hired experienced executives to head Development, Consulting, and Sales. But by late 1994, Liemandt was not happy with the way things were going. Things were good at present, but he worried about the future:

Source: Austin (1998, pp. 2-3).

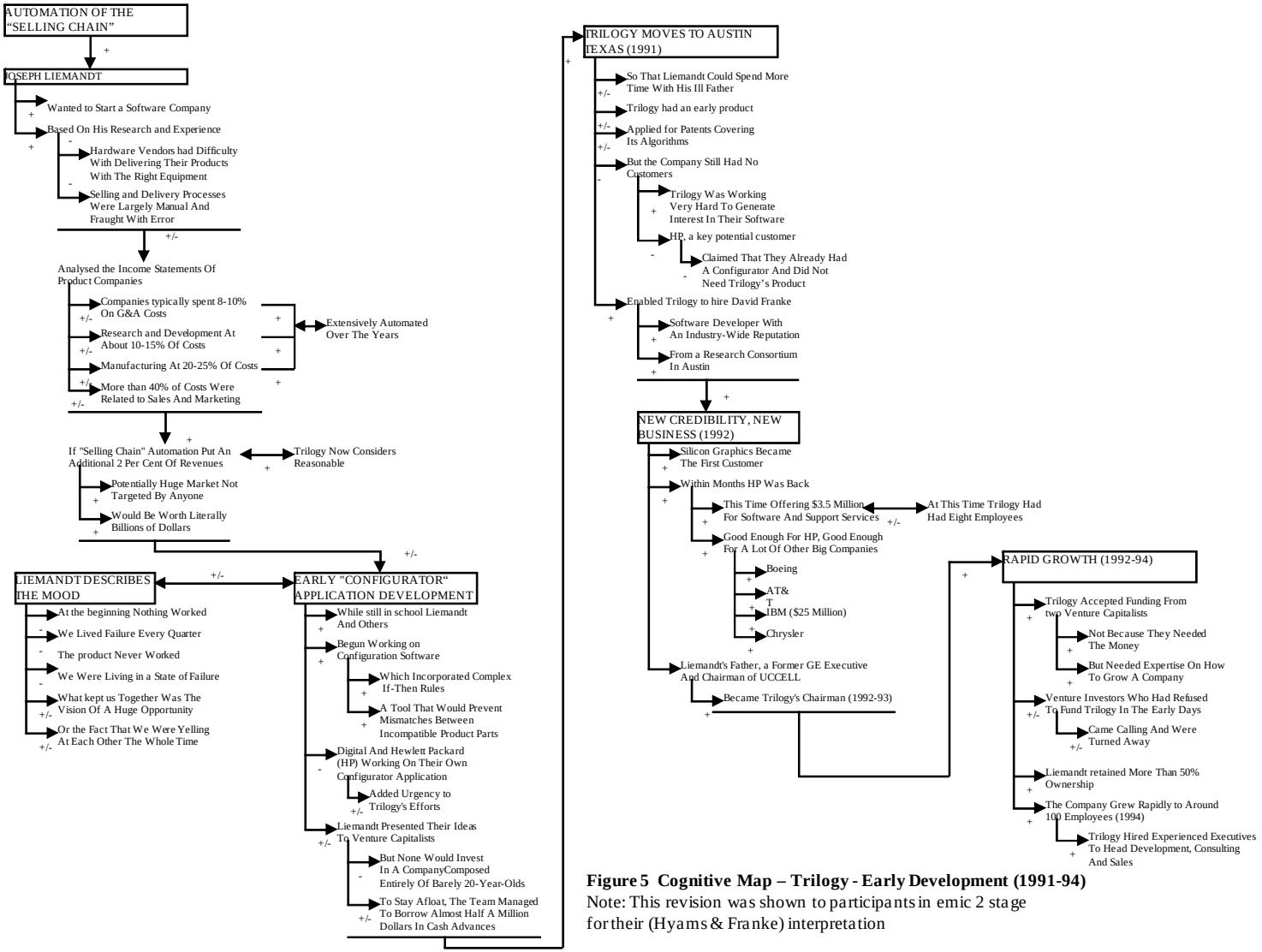


Figure 5 Cognitive Map – Trilogy - Early Development (1991-94)
Note: This revision was shown to participants in emic 2 stage for their (Hyams & Franke) interpretation

While studying at Stanford University, Joseph Liemandt wanted to start up a software company. He had unearthed an opportunity where computer hardware vendors had difficulty delivering products with the correct parts and equipment. Their selling and delivery processes were mainly manual, and were error-prone. Liemandt conducted further analysis on the income statements of selected computer companies, finding a breakdown of costs into the following percentages:

- General and Administration (G&A) (8-10%)
- Research and Development (R&D) (10-15%)
- Manufacturing (20-25%)
- Sales and Marketing (40%).

G&A, R&D and Manufacturing had been extensively automated, but by 1990 there was very little automation within sales and marketing functions.

Liemandt calculated that attempting to automate aspects of the "Selling Chain" (the sales and marketing functions) to enable an additional 2% of revenues would be potentially worth billions of dollars – and was a new potential market not currently targeted by anyone. Austin (1998) indicated as an additional note that the estimate of adding 2% to revenues was a figure considered as reasonable by Trilogy. Liemandt and some other students at Stanford started work on developing configuration software. Configuration software enables the user to incorporate all essential features and equipment for a product, plus nomination of optional additional products and accessories.

In the late 1980s, most quotes for computer hardware were line-by-line items manually entered without any form of validation or checking. For complex large system quotes, such as for multimillion dollar tenders, hardware consultants were employed to check system configurations and line-by-line. The author of this dissertation was a specialist consultant in the area of the configuration and pricing of complex computer systems in the late 1980s.

Early configuration systems were developed by DEC and HP in the mid-to-late 1980s, but they both required mainframe computer power to run through complex if-then rules, and even then only standardized systems with few additional options could be processed into quotes in a reasonable turn-around time. Nevertheless, as minicomputer and small server performance was rapidly increasing, these configuration systems had the potential to be placed on smaller very powerful UNIX-based systems emerging at that time, and eventually on Windows-based PCs.

Liemandt realized that Trilogy needed to bring its configuration software to market quickly to counter DEC and Hewlett-Packard's early lead. Decision-making at this stage was all about evaluating options to commercialize and bring to an application rapidly to market. Trilogy needed capital to accelerate application development. Liemandt sought venture capital support but no venture capitalist was prepared to invest in a company with such a young group of developers. However, Trilogy's founders had such a strong faith in its capability and vision for its application that they leveraged its own credit cards with cash advances of almost half a million dollars. Such an attitude to risk-taking, that is, risk enough money to hurt, but not to break the bank is a fundamental organizational value at Trilogy and is explored in more detail in other sections of Austin's (1998) case study.

Liemandt described the atmosphere of Trilogy at the time (1990-91) as a state of failure where nothing worked, but a strong vision kept the team together. Or the team seemed to stay together by "yelling at each other the whole time". A "circuit-breaker" for Trilogy seems to be Liemandt's decision to move Trilogy to Austin, Texas, in 1991. Liemandt wanted to spend more time with his ill father in Austin and moved there prior to his graduation from Stanford. Other members of the Trilogy gradually moved to Austin, some before graduation, some after graduation.

By the time the move to Austin was completed, Trilogy had developed an early application. In an unusual move for software developers, Trilogy applied early for patents – not so much for the actual application but for the algorithms used in the if-then rules within the application. Although Trilogy was trying hard to generate interest in its application, they had no customers. In a twist suggesting Trilogy knew that its application was superior to Hewlett-

Packard's existing configuration software, Trilogy was actively targeting Hewlett-Packard as a potential customer. At the time, Hewlett-Packard rejected Trilogy's overtures, claiming that it already had its own configuration software.

Moving Trilogy to Austin enabled Liemandt to hire David Franke from a research consortium in Austin in 1992. Franke was a well-known software developer with strong links to various hardware and software suppliers and he gave Trilogy a substantial boost in industry credibility. Trilogy's increased industry credibility through Franke translated into new customers. Silicon Graphics (SGI) became Trilogy's first customer, and within a few months Hewlett-Packard signed up in a $3.5 million deal for software and support services. Trilogy was still a small start-up company with eight employees, facing exponential demand for its application.

Securing Hewlett-Packard as a key customer appeared to legitimize Trilogy in the eyes of several large companies who soon also signed up with Trilogy – and for large contracts. Boeing, AT&T and Chrysler rapidly signed up, while IBM took on Trilogy's software and support for over $25 million. Liemandt's father Greg, although diagnosed with a fatal illness, was appointed Chairman of Trilogy during 1992-93, sharing valuable experience and knowledge from senior executive roles at GE and being Chairman at UCCELL. Trilogy now needed knowledge and resources to turn into a large enterprise servicing large corporate customers.

Trilogy was seen in this period as a great investment opportunity by several venture capitalists, but most were turned away as Liemandt really wanted expertise and resources to rapidly grow the company, and not just capital. Liemandt accepted funding from two venture capitalists that provided these additional benefits, but maintained 50% ownership of the company. Trilogy grew rapidly through 1992-94 to around 100 employees and experienced executives were hired to establish organizational structure for development, consulting and sales. Trilogy was growing strongly with a "killer application" for sales & marketing – and that was all before the Internet revolution arrived.

Updated Cognitive Map 1 – Early Development (1991-94)

The cognitive map was revised through direct editing in PowerPoint with inputs from interviews with Franke (2002) and Hyams (2002). Several sections of the original map were validated with minimal changes, but there were some significant additional insights for other sections. Figure 6 presents the updated map.

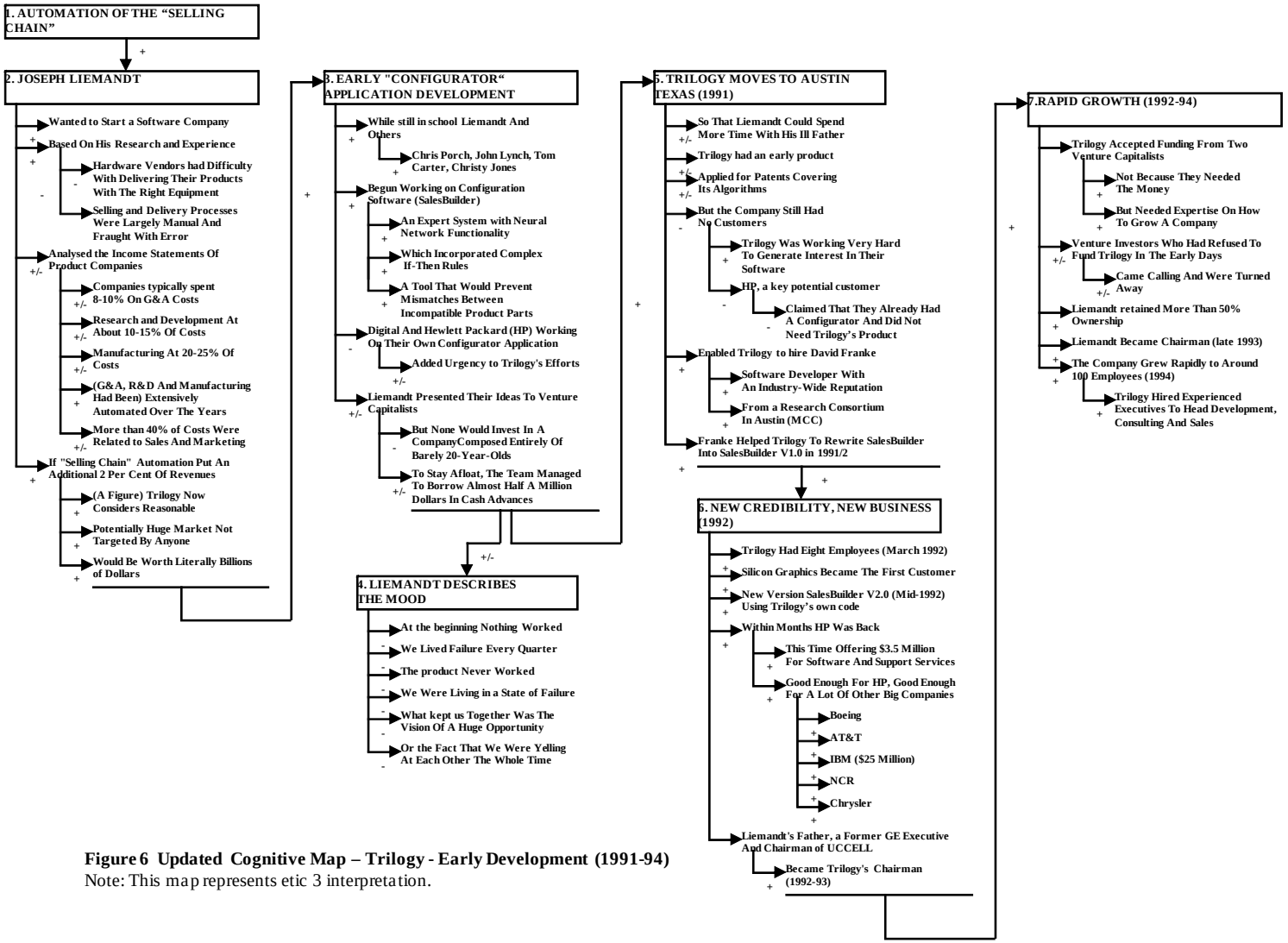


Figure 6 Updated Cognitive Map – Trilogy - Early Development (1991-94)
Note: This map represents etic 3 interpretation.

Sections 1 and 2, apart from some formatting adjustments, were validated unchanged from the original cognitive map. Section 3, on "early configurator application development", was updated to indicate that apart from Liemandt, the other early developers were Chris Porch, John Lynch, Tom Carter and Christy Jones. The actual configuration application was known as SalesBuilder and was based on an expert system with neural network capability. These features enabled the setting up of complex rules for selection of features and options.

Section 4 was validated unchanged from the original map. Trilogy's move to Austin covered in Section 5, was amplified to provide more insight into Franke's move to Trilogy. Franke had worked with MCC in Austin, a company that was focused on developing new computing technologies and software. Franke had substantial knowledge in software development, expert systems and he was already quite familiar with algorithms and developments related to configuration applications. Franke helped Trilogy formalize SalesBuilder into a formal release SalesBuilder V1.0.

Section 6, on New Credibility and New Business, was updated to incorporate additional insights. Trilogy had eight employees in March 1992. Trilogy developed SalesBuilder V2.0 in mid-1992, using its own code and developers. NCR was identified an additional large new customer in 1992. Section 7, covering Rapid Growth, was validated unchanged except that Joseph Liemandt became chairman in late 1993 (after his father Greg).

Cognitive Map 2 – Industry and Competition

The cognitive map was developed from the "Industry and Competition" section in Austin (1998, pp. 4-5 – see Exhibit 6. Figure 7 presents this cognitive map. The map provides a cognitive description of Trilogy's perception of its competition and industry space from the early 1990s through to 1999.

Exhibit 6 Text Extract - Trilogy – Industry And Competition

Industry and Competition

Liemandt's original analysis of spending patterns versus degree of automation in computer product firms had identified a wide-open market worth at least $10 billion. The few companies that were in that market at the time were bit players, selling things like contact management software for salespeople. Most of the functionality that constitutes the bulk of the "selling chain"-catalog updating, configuration, pricing, bid preparation, commission calculation-was performed manually or by software written by product firms themselves. Trilogy had pushed rapidly and successfully into this mostly empty space.

But Trilogy's success did not go unnoticed. Beginning in about 1993, new companies like Aurum, Brightware, Calico, Clarify, Remedy, Scopus, Siebel Systems, and Vantive entered the general area of sales and marketing automation. Some of these companies targeted niches that were not in immediate competition with Trilogy. But all were operating in the same general space, going after that 40-plus percent of P&L spending that Liemandt had first noticed as a student. More worrying than these small players, however, was the awakening interest of the giants of Enterprise Resource Planning (ERP)-rapidly growing companies that were many times larger than Trilogy, such as SAP, Oracle, Peoplesoft, and Baan (see Exhibit 3 for profiles of these companies).

Trilogy had intentionally positioned itself as an "enterprise software" company, meaning that its products were designed to work together to provide end-to-end functionality for a major segment of a customer's business (the selling chain). This was necessary because Trilogy's corporate customers were increasingly looking to build or buy integrated systems. Companies that did not sell enterprise products risked losing out to companies with more integrated and broader product offerings. But ERP vendors saw the enterprise software market as their turf. One company's supply chain, reasoned the ERP giants, was another company's selling chain. As the experts on integrating a customer's "back office"-the value stream from procurement through production to delivery-it seemed only natural to the ERP vendors that they should also integrate the "front office"-the selling chain. Liemandt summarized the threat to his company in stark terms:

They decided that Trilogy had done some fantastic research for them and that they'd just come in and take it over. The question was (and still is), "can we withstand the onslaught of giants ten times as big who want to move into our space?"

As early as 1993, Trilogy had realized that the number one threat to its long-term well-being was SAP, the largest of the ERP vendors, which by 1998 owned 70 percent of the back office automation business for Fortune 500 customers. In 1997 and 1998, the threat from SAP and the other ERP vendors became more immediate. Baan purchased Aurum. Peoplesoft announced partnerships with Vantive and Siebel Systems (which itself bought Scopus). SAP pointedly failed to invite Trilogy to exhibit at Sapphire 1998, the SAP-sponsored tradeshow for its own customers, even though the company had

participated in earlier years.4 At that same tradeshow, Hasso Plattner, SAP's chairman and cofounder, announced to his customers that 80% of the company's R&D going forward would be aimed at building front office products.

Trilogy had a considerable head start on ERP vendors in the development of key technologies, especially configuration software, some of which was by then protected by patents. But the protection provided by patents would be short-lived. Whether Trilogy would remain a factor would depend far less on past accomplishments than on what they could accomplish in the future.

Source: Austin (1998, 4-5).

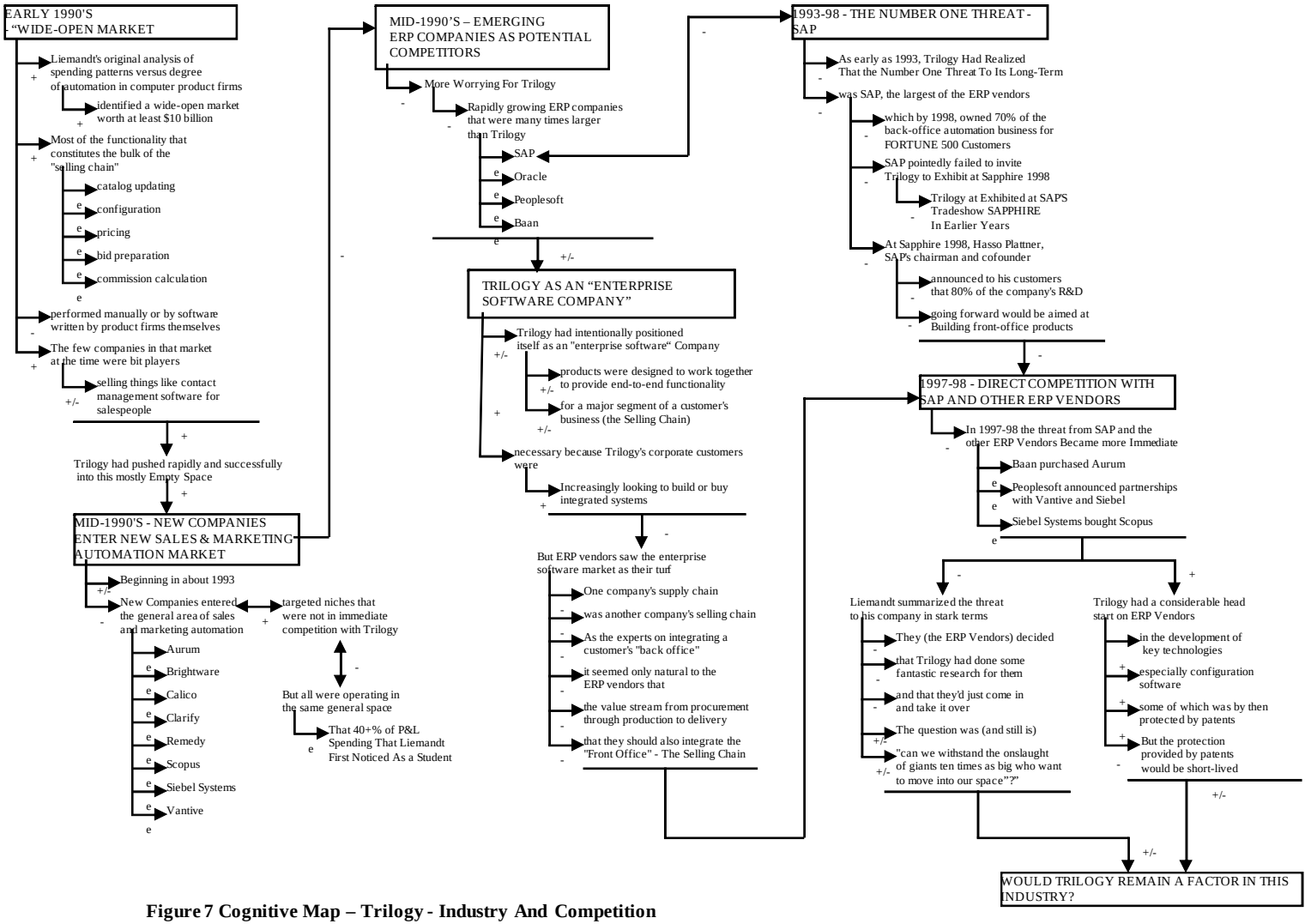


Figure 7 Cognitive Map – Trilogy - Industry And Competition

Note: This revision was shown to participants in emic 2 stage for their (Hyams & Franke) interpretation.

Liemandt identified an unmet market opportunity (see also first cognitive map) that he estimated to be at least $10 billion in the area of the sales & marketing functions, or the "front-end" or "front-office" or the "selling chain".

Most activity in the selling chain activities for computer hardware companies such as catalogue updating, configuration, pricing, bid preparation and sales/consulting commission calculations were processed manually. There were a few companies that sold software for specific activities in the selling chain such as contact management software. There were some forms of simple customer database applications available for direct marketing programs but CRM as a concept emerged in the late 1990s.

Trilogy was one of the first companies with a solution that went further than just one component of the "Selling Chain". However, it was not long before Trilogy faced competition from several fronts. Several of the companies that we would now see as CRM vendors were created in the early 1990s. Eight companies are noted in the original cognitive map, of which probably Siebel Systems is best known for sales force management applications.

Most of the companies noted were not in immediate competition with Trilogy – but they all could be classified as competition, depending on how an industry space for a "selling chain" was defined. Trilogy faced more aggressive and powerful competition from a different front. Trilogy's executive always saw the company as an "enterprise software" company that could address a full end-to-end "selling chain". Trilogy was in part responding to customers who wanted to build or buy systems that at least integrated the selling chain. Such positioning put Trilogy directly in the path of large and rapidly growing enterprise resource planning (ERP) vendors. Companies such as SAP and Oracle had been established at least 10 years longer than Trilogy and had grown rapidly through the 1980s and 1990s offering various key enterprise applications.

SAP had grown out of offering enterprise financial software, expanding into manufacturing and logistics in the early 1990s. SAP could be put on several computer hardware platforms but was often used on IBM systems through the 1980s and early 1990s. Oracle rode the Open Systems/UNIX boom of the late 1980s and early 1990s to be the major database application for various enterprise applications to run on UNIX hardware vendors including Sun, Hewlett-Packard, and DEC's UNIX systems. Oracle was also early to position its applications as e-business solutions to work with the World-Wide Web (working with such applications as NetDynamics – see the NetDynamics case study). Peoplesoft was created in the late 1980s with Human Resource Management (HRM) applications but spread quickly into other enterprise applications, working closely with UNIX computing vendors, but also with IBM and DEC's VMS systems of the early 1990s. Baan was a Dutch company that emerged through enterprise financial applications in the early 1990s.

The ERP companies' main focus was integration of "back-office" activities such as manufacturing, logistics, and financial management. However, it was logical for the ERP companies to expand its application suites right from procurement to delivery, including integration of front-office activities. Trilogy identified SAP as its number one long-term competitor as early as 1993. SAP dominated the "back-office" ERP business with about 70% share of that business by 1999. However, for about five years Trilogy worked with SAP as a key ISV and regularly exhibited at Trilogy's main tradeshow, SAPPHIRE.

SAP changed its strategy in 1998 to focus heavily in building front-office applications. One side-effect of that strategy was not to invite Trilogy to SAPPHIRE 1999. At that event SAP's Chairman committed that 80% of SAP's R&D would go to development of front-office applications. By 1997, competition between the main ERP vendors and Trilogy was quite direct, as a number of ERP companies either acquired ISVs with front-office applications, or they attempted to develop its own applications.

Trilogy's management believed that the company possessed a considerable lead over the ERP vendors in the selling-chain area. Trilogy was very strong in configuration technology, for which it held some significant patents. However, Trilogy needed to continually develop new front-office-related applications and grow fast enough to gain sufficient critical mass to take on much larger companies pushing into the selling-chain space. Liemandt directly questioned

whether Trilogy could survive as a key player at least in the selling chain industry that he had defined back in the early 1990s.

Liemandt was reflecting on a not uncommon problem for small start-up companies with strong new technologies facing up to larger more established companies attempting to break into its area of expertise. Other case-studies in this project and particularly those with enterprise software solutions have faced the same problem as Trilogy's.

NetDynamics gained a strong early lead with its application to link WebPages with databases, but was then acquired by Sun, as it needed size and resources to support a rapidly growing enterprise customer base. Kana rapidly acquired ISVs to develop a full e-CRM solution. It moved from a focused application on e-mail communication management to a redefined and more integrated e-CRM and e-commerce applications space.

Trilogy could see that they needed to at least strengthen its product offering for the selling chain, although at the same time, it was not committing to redefine its offerings to match the total ERP solutions. The overall cognitive map presents Trilogy's management perceptions of competitive evolution through the 1990s. The map also highlights Trilogy's executive team's strategic mapping of its competitive position, initially in a new industry, but then in a more broad and redefined industry with much larger and more aggressive competitors on the same perceived turf.

Updated Cognitive Map 2 – Industry and Competition

The cognitive map was revised subsequent to interviews with Franke (2002) and Hyams (2002), see Figure 8. Most of the original cognitive map was validated unchanged, but there were some additional insights and structural updates incorporated into the map.

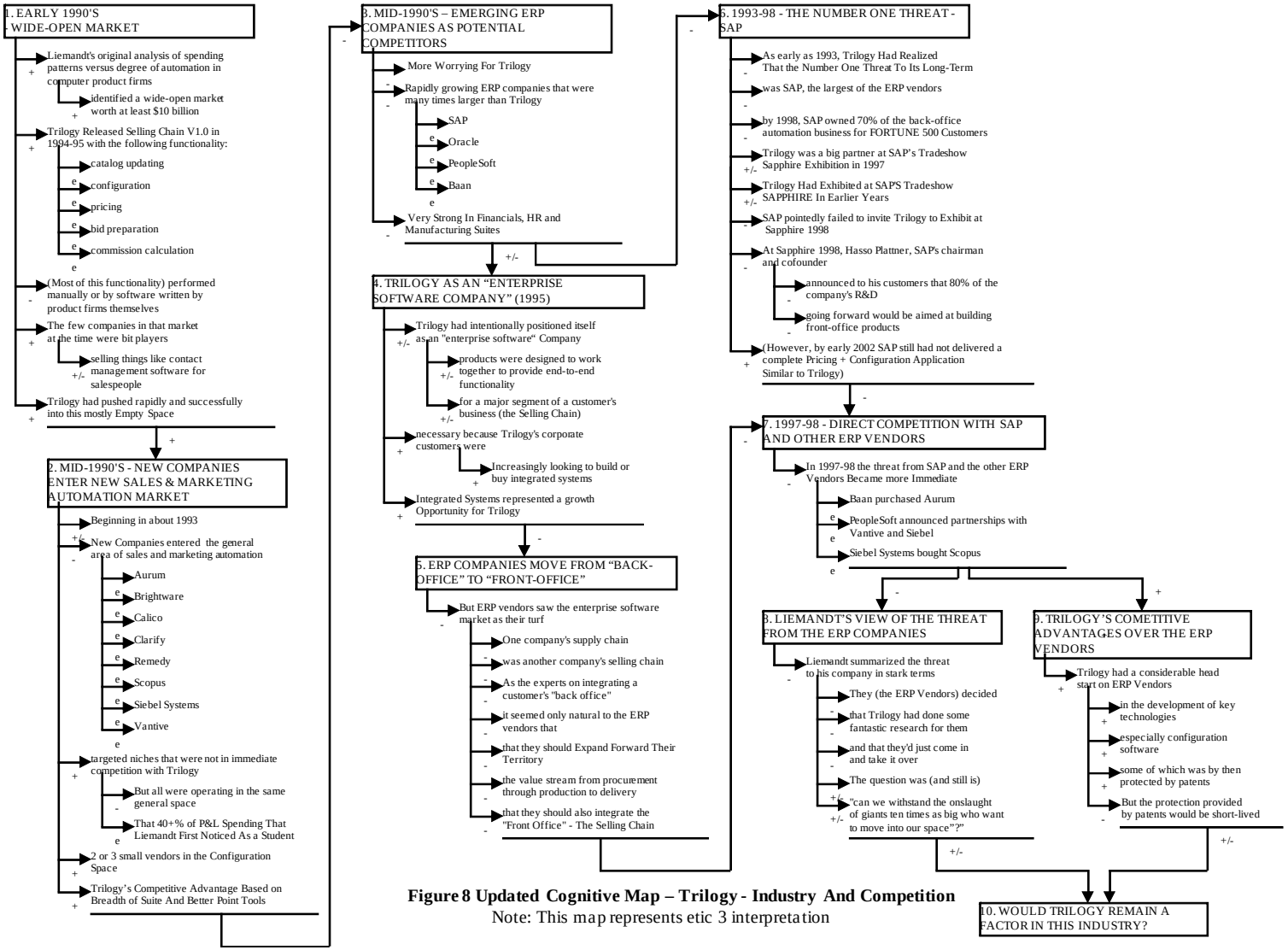


Figure 8 Updated Cognitive Map – Trilogy - Industry And Competition
Note: This map represents etic 3 interpretation

The first section of the map covering "Early 1990s – Wide-open Market" (box 1) was validated with only two modifications. The first indicated that Trilogy released an application Selling Chain V1.0 in 1994-95 with the functionality described in the original map. The second modification was the incorporation of the comment in the original map relating to Trilogy's moving into a mostly empty space, into the first section of the updated map. The second section, "Mid-1990s – New Companies Enter New Sales & Marketing Automation Market", was validated with two modifications related to the same point. Franke (2002) noted that there were two or three vendors in the configuration space. Trilogy's competitive advantage was based on the breadth of its new application suite (Selling Chain V1.0) and better point-by-point configuration capability. The third section on emerging ERP companies as potential competitors was validated unmodified, but extended to highlight the ERP vendors' strengths in Financials, Human Resources Management and Manufacturing suites.

Trilogy's positioning as an "Enterprise Software Company" (box 4) was validated without modifications to content, except that the sub-section on ERP vendors seeing enterprise software as its turf was reformatted into a separate section (box 5 – ERP Companies Move From "Back-Office" to "front-office"). The original section on SAP being the Number One threat to Trilogy was slightly reformatted (See box 6) but content was validated without modifications. An additional point was incorporated into the section indicating that, even by early 2002, SAP had still not delivered a complete Pricing and Configuration application similar to Trilogy.

The section on the consolidation of ERP Vendors with other "front-end" vendors through selected acquisitions was validated unchanged for content. However, this section, including the subsections on Liemandt's view on the threat of from the ERP companies, and Trilogy's competitive advantages over them, were split into three sections. (See box 7 for Direct Competition, box 8 for Liemandt's view of the ERP companies, and box 9 for Trilogy's Competitive Advantages over the ERP Vendors).

The question at the end of the original map regarding whether or not Trilogy would remain a factor in the industry was reformatted as box 10. The revised cognitive map largely validates the original map, with some extra insights specific to Trilogy's actual Selling Chain application suite.

Cognitive Map 3 – Software Development

The cognitive map was developed from the "Developing the Product" section, Austin (1998, pp. 7-8 – see Exhibit 7). The cognitive map is presented in Figure 9. The map provides further insights into Trilogy's product development capability and its fundamental philosophies of software development.

Exhibit 7 Text Extract - Trilogy – Software Development

Developing the Product

Trilogy's marketing goals depended vitally on the company's product development capability. Specifically, marketing objectives required that developers sustain the competitive advantage Trilogy enjoyed in configurator technology while dramatically broadening the product to fill the enterprise needs of large and small customers in a variety of industries.

Trilogy aspired to maintain a software development capability that was second to none. Their comparison set for evaluating themselves in this area was not their direct competitors, but other world class development organizations, especially Microsoft. In their aim to be the best, they believed that they had largely succeeded. Liemandt was convinced that no other enterprise software vendor was even a close second to Trilogy in development capability. Scott Snyder, Trilogy's Senior Development VP, estimated that 15 or 20 successful software companies could be built around the talent in Trilogy's development organization.

Central to the company's development capability was the "rule of the super coders," which held that one superstar programmer could do the work of ten average programmers. "Getting the most out of great developers," remarked Snyder, "is one of the things Trilogy does amazingly well." The development process was geared toward giving Trilogy's superstar programmers the support and freedom they needed to produce great products. Snyder described some of the company's fundamental philosophies of software development:

Our development is based around four basic philosophies. Small teams, very small from a traditional development standpoint. We expect entire new products to be created by one or two superstar programmers you can count on to deliver great products quickly. Complete ownership of the product at the developer level from initial product requirements gathering through product support. We don't have a separate change team that insulates the developers from the impact of producing poor quality products. Intense focus on automation in order to free the developers (or anyone else) from having to spend their time manually performing frequently repeated tasks like regression tests. Finally, a focus on incremental development model that allows us to deliver new functionality quickly and provides us the flexibility to react to changes in the market quickly.

A key feature of the development process was that it evolved to maintain responsiveness to the market, becoming more structured as the product matured. As the product grew beyond a certain size, explained Snyder, maintaining responsiveness and high product quality depended on some key disciplines:

The goal is to maintain your code at ship level quality on a weekly basis. When a developer drops code for a new feature or bug fix into the build, it must be accompanied by the appropriate suite of automated tests to validate that the changes work as expected. These tests are added to the existing suite and the entire set is executed every time the product is built, whether that was a weekly or nightly build.

Exhibit 7 Text Extract - Trilogy – Software Development (Cont'd.)

If you had to develop a feature that took longer than the weekly build cycle, you branch your development, develop the feature, develop tests, merge it all back into the main build, then rerun all the tests on the integrated code. In addition to the individual product tests, we also have automated system tests which test the interactions between products. The goal is to constantly improve the quality of the product as you increase its functionality. Again, a very incremental model. It's awesome and brilliantly suited for an environment which requires you to react quickly to any new requirement or change in market direction as long as you maintain the quality discipline.

As the product grew in complexity, programmers retained absolute freedom to add features in whatever way they saw fit, but they were obliged to maintain quality. As interdependencies developed between different developers' programs, the build and release cycle became more structured, with decisions being made about the timing of the release of new features on a feature-by-feature basis. New features were scheduled around a plan that used 60 to 70 percent of Trilogy's development capacity with the remaining 30 to 40 percent held in reserve for late breaking and urgent fixes.

Source: Austin (1998, pp. 7-8).

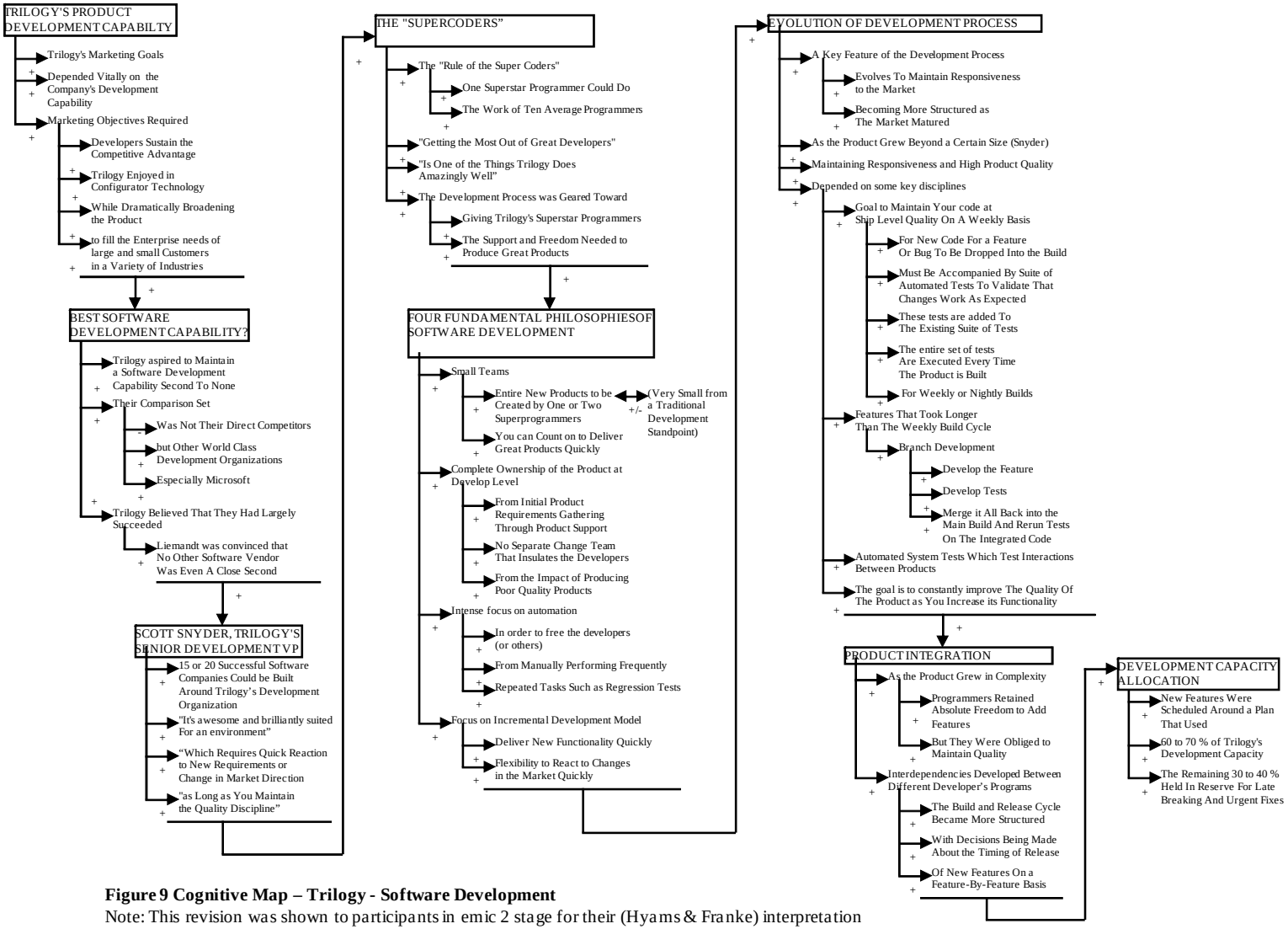


Figure 9 Cognitive Map – Trilogy - Software Development
Note: This revision was shown to participants in emic 2 stage for their (Hyams & Franke) interpretation

According to Austin (1998), Trilogy‘s marketing objectives focused heavily on the company’s development capability. Trilogy’s developers had the goal of sustaining its competitive advantage in configuration software, but also to quickly broaden the product to meet the enterprise needs of a wide range of customers. Balancing leadership in their core competency with broadening the appeal of the product to more customers was a logical approach to the rapidly growing enterprise software market during the mid-1990s. Such an approach also required the rapid recruitment of large numbers of very highly skilled application programmers.

Trilogy was created as a company with a deep belief in aspiring to be the best software development company in the world. Trilogy initially benchmarked its emerging development capability against other world-class development organizations, but particularly Microsoft. By 1998, Liemandt was convinced that Trilogy was the best by a long way in software development capability. Trilogy’s Senior Development Vice-President in 1998, Scott Snyder provided insights into Trilogy’s view of itself as a best-in-class software development company. He regarded Trilogy’s development organization as “awesome and brilliantly suited for an environment which requires quick reaction to new requirements or change in market direction, as long as you maintain the Quality Discipline” (Austin 1998, p. 8). Snyder also believed that Trilogy’s development organization could be the basis for “15 or 20 successful software companies”.

In order to grow and maintain a best-in-class software development capability, Trilogy encouraged a concept of “Super Coders”, whereby one Superstar Programmer could do the work of ten average programmers. Trilogy focused on “getting the most out of great developers” through “giving them the support and freedom needed to produce great products”. A focus on maximizing productivity from high-quality programmers, right through to linking its achievements through to marketing goals, is an unusual feature of Trilogy as an organization.

Trilogy’s software development was built around four fundamental philosophies which Austin has recounted from Scott Snyder. Small teams, in fact teams of even one or two Superprogrammers developing entire applications (or at least modules of applications), follows logically from the very high value Trilogy placed on these individuals, both from a productivity and quality output perspective.

Complete ownership of the product at developer level is driven down to a unit of analysis of a team of one or two Superprogrammers. Trilogy gives its teams control of the full application development cycle – both for good and poor quality products. In reality the ownership is at a collective team level and this is evident where there are requirements to create new modules or new applications. Intense focus on automation enabled Superprogrammers to be freed up from manually performing frequent repetitive tasks and to focus on direct application development. Focus on incremental development model seems initially to be a contradiction in that Superprogrammers might be expected to quickly create radical new applications and products. However, Trilogy seems to have placed a higher priority on sufficient modularization and break-down of development components to enable the Superprogrammers to quickly change components in response to fast changes in markets.

Applying the four fundamental philosophies of software development to actual development processes required balancing the need to maintain responsiveness to markets with becoming more structured as markets matured. According to Tom Snyder, maintaining this balance depended on some programming and development disciplines. Development teams should have a goal to maintain its actual code at a level where it could be shipped (or be certified as completed) on a weekly basis. A new component or feature or bug that was to be incorporated into the weekly build of the overall application had to include a set of automated tests that would validate that the addition would work as expected – outside and within the weekly build.

The tests would be executed every time a build was undertaken. While a typical build may be undertaken weekly, some applications were built into new versions nightly – with the tests having to run at the frequency of the build. Development Teams had to quickly judge if a feature or component or bug was likely to take longer than one weekly build cycle. Anything

longer than the weekly build cycle prompted a branching of development, probably to a new or revised small team, which would undertake the development with the automated tests and then ensure that it merged successfully back into the overall application, usually in the next weekly build. Additional automated systems tests were created to test interactions between applications, modules and other Trilogy products. Trilogy focused on constantly increasing the quality of its overall product-line as it also increased functionality within the line.

Over time Trilogy's product grew in complexity and, while the Superprogrammers retained the freedom to add features, they had to maintain overall product quality. Interdependencies emerged across and between the development teams with more decision-making required across the teams.

Cross-team decision-makers contributed to increasingly structured build and release cycles, and all additional features were subjected to substantial assessment before being accepted for both development and incorporation in the product. By the late 1990s, Trilogy's development processes had evolved to allocate 60 to 70% of Trilogy's development capacity for new features, and 20 to 30% in readiness for late breaking and urgent fixes. The cognitive map captures insights from a company that puts the very highest priority on best-in-class software development. There is literally a unique definition – and acclaim – of a "Supercoder" or "Superprogrammer" that is capable of much higher application development productivity than typical programmers. Trilogy pushes this elite group of programmers further with philosophies and supporting processes designed to maximize its productivity. Trilogy appears to have achieved scalability throughout the late 1990s, with its "Superprogrammer" approach, maintaining frequent releases and updates, with both high-quality shipment code and effective bug and quality fix processes.

Updated Cognitive Map 3 – Software Development

The cognitive map was validated almost unchanged through interviews with Franke (2002) and Hyams (2002), and is presented in Figure 10. The original cognitive map was validated unchanged except for the numbering of sections and for a comment from Franke (2002) that there were 20-30 small development teams during the late 1990s (this comment was added into the *Small Teams* subsection of "Four Fundamental Philosophies of Software Development").

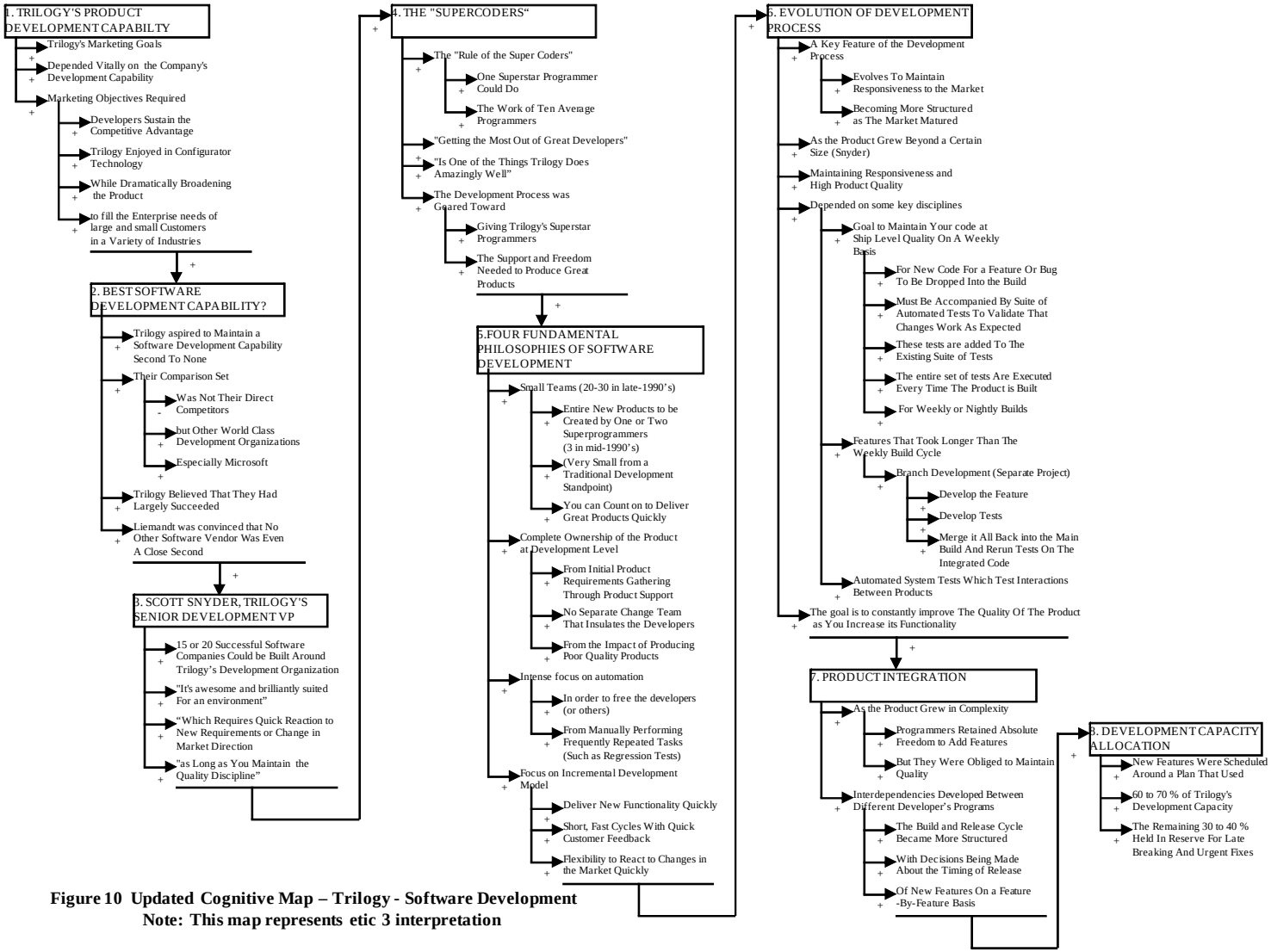


Figure 10 Updated Cognitive Map – Trilogy - Software Development
Note: This map represents etic 3 interpretation

Franke (2002) and Hyams (2002) agreed that Austin's (1998) account of Trilogy's software development was accurate and with substantial insights – and these had been effectively captured in the original cognitive map.

9.6.4 New Cognitive Map 4 – Industry-focused Application Development (2000-01)

During the interviews for the Trilogy project, suggestions were put forward that both a new DSA model and supporting cognitive map for Trilogy's application development in 2001 be developed for the case study. The cognitive map was developed from a Subset from the "Trilogy Goes Vertical – Industry-focused Enterprise Solutions" Vignette (Vignette was presented in Exhibit 3). The subset is presented in Exhibit 8 and Figure 11 presents the cognitive map output.

Exhibit 8 Extract From Vignette - Trilogy's Industry-Focused Application Development

TRILOGY GOES VERTICAL – INDUSTRY-FOCUSED ENTERPRISE SOLUTIONS

In 2000, Trilogy restructured from a horizontally focused application vendor toward using a Software Development Methodology designed to quickly conceptualise, create and deliver highly verticalized applications. Previously, from 1993-1999, Trilogy's application development strategies were based on Trilogy Product Roadmaps.

The Executive Team established an Operations Group with five vertical divisions focused on key enterprise customers and products:
* Automotive
* Computer
* Telecommunications
* Financial Services
* New Business

Trilogy has a two-dimensional matrix of management and delivery resources, located in Core company-wide Groups and within each Division with deep industry experience. These resources cover:
* General Managers plus Functional Vice-Presidents
* Development Staff (Division-Specific)
* Consulting
* Human Resources
* Finance
* Solutions Marketing and Business Development
* Product Management and Presales Resources

Trilogy's Fast Cycle Time (FCT) Software Development Methodology is now based on delivering vertical applications within Industry Maps set up for each division for a two-year timeframe. There are four phases in this methodology:

1. Product Ideation
During this phase, Trilogy's Division Managers with their Solutions Marketing and Product Marketing teams generate ideas for products, creates a vision for the product and the Industry, applies an investment justification process to the vision and product, and develops an Industry Roadmap supported by a Business Case. All product conceptualization and development is driven with a vertical industry focus.

Exhibit 8 Extract From Vignette - Trilogy's Industry-Focused Application Development (Cont'd.)

2. Product Planning
During this phase, the capabilities of the product will be identified and sequenced for delivery. Product features will then be expanded into actual development schedules including actual release features and initial estimates at release dates. During this phase, Product Management takes over with Development Teams in driving product planning.

3. Product Development
This phase consists of gathering the detailed requirements for the product, and commencing development using Trilogy's Fast Cycle Time methodology. Although Product Management and Development Teams are driving the product development process, chartered customers associated with specific Trilogy divisions may be involved in testing early versions of the product.

4. Product Delivery
Product Delivery is defined as beyond Beta Release. There is extensive interaction between Product Management, Development Teams, charted customers, and solutions marketing groups. The final product will be delivered during this phase. Once in production, the product will be maintained with patches, bug fixes and sub-releases. Consultants may work with chartered customers in deployment of new products.

Source: Extracted From Exhibit 6.

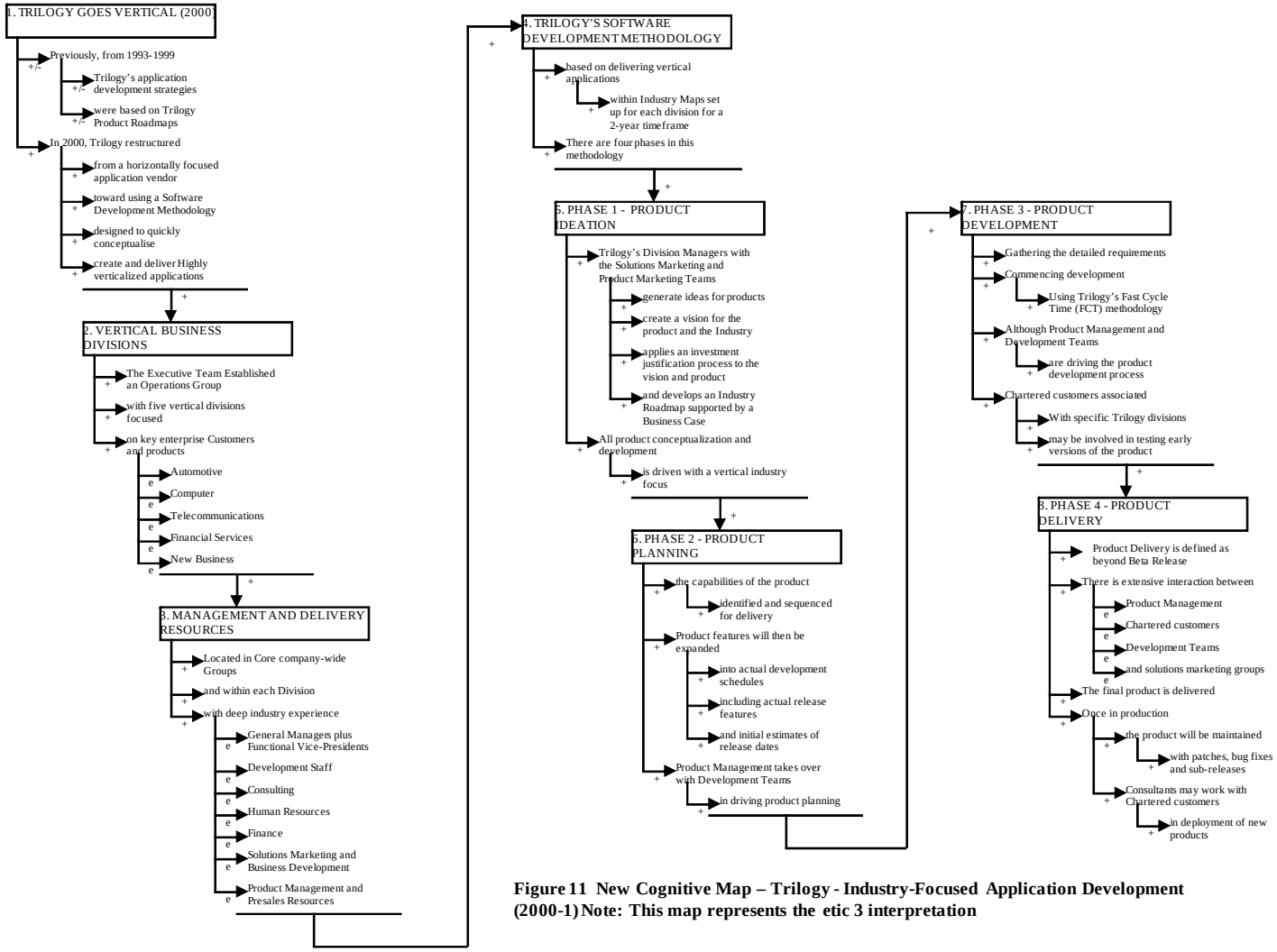


Figure 11 New Cognitive Map – Trilogy - Industry-Focused Application Development (2000-1) Note: This map represents the etic 3 interpretation

From 1993-99, Trilogy's focus was on excellence in software development, mainly in horizontal configuration applications. Trilogy had been servicing enterprise customers since its inception due to the nature of its configuration application and extensive potential for automation of key sales and marketing activities using the application and extended enterprise-oriented versions of it. However, in 2000 during and after the "dot.com" crash and falls in technology/computing stocks, the company restructured toward a focus on selected vertical industry sectors (see box 1).

Trilogy revamped its software development methodology toward quickly conceptualizing and road-mapping applications against targeted vertical markets, thereby developing and delivering highly tailored applications for specific industries and customers. Trilogy's Executive Team established an Operations Group with five vertical divisions, based on its key enterprise customers and products (box 2). Trilogy's five divisions are Automotive, Computer, Telecommunications, Financial Services, and a New Business Division.

Each division was set up with a complete set of management and delivery resources (box 3). Each division possessed resources with deep vertical industry experience in all management areas, including General Management, Development, Consulting, Human Resources, Finance, Solutions Marketing and Business Development, plus Product Management. In terms of decision-making, Trilogy's restructure was a radical departure from decision-making focused at the Superprogrammer or at the small development team level, toward managers with deep industry experience – but not necessarily focused on software development.

Trilogy's software methodology is now based on delivering vertical applications within each Division (box 4). Each Division has its own industry maps set up for a two-year timeframe which is designed around a four-phase development methodology. The first phase of software development methodology is "Product Ideation" where Division managers, with solutions marketing and product marketing teams, work together on generating ideas and vision, which in turn are translated into products within the context of targeted industries.

Although there appears to be substantial freedom to visualize and conceptualize new applications in this early phase development, there is an immediate requirement to apply an investment justification process and to immediately map them to an industry roadmap with a business case. All product conceptualization and development is undertaken with direct alignment to specific divisional industry roadmaps, and thus is driven right from the outset with a vertical industry focus.

In the second phase of development, Product Planning, Product Management, with Development Teams (within each Division), drive the definition of product capabilities and scheduling of development and delivery (box 6). Product features are identified and defined into full development schedules with estimated and declared release dates. In the third phase of development, Product Development, Trilogy employs a proprietary application development methodology (Fast Cycle Time, or FCT), to quickly develop new features and applications (box 7).

Although the full four phase methodology is known as FCT, the actual product development phase is an industry-focused evolution of Trilogy's earlier software development methodology, using small teams of Supercoders with automated tests, as described in the cognitive map on software development.

Key customers (or chartered customers) for specific Trilogy divisions may be involved in the testing of early versions of a product. These customers may also have had some say in product ideation via consultants working with them and through to the Solutions Marketing Group. Trilogy commenced the development of Internet-enabled applications in 1996-97 mainly using Java, C++ and new object-oriented developer applications, plus they were early adopters of XML. Trilogy's configuration systems saw am almost immediate deployment into online ordering systems in 1996-97 and have become one of the major enterprise-wide e-commerce applications. These applications are now growing strongly in all of Trilogy's targeted industry segments through further customization and incorporation of new features and e-business technologies.

The final development phase, *Product Delivery*, commences when the product is beyond beta Release (box 8). Prior to final delivery, there is substantial collective interaction between product management, chartered customers, development teams and solutions marketing groups. A formal launch and roll-out of the product is finalized and the product is launched and delivered.

Ongoing product support and maintenance are classified as part of product delivery. Patches, bug fixes and sub-releases are planned and delivered within the product delivery phase. Further customization and support may be required through consultants working with customers on the deployment and installation of the new product. The cognitive map on industry-focused application development captures a radical transformation in decision-making associated with application development at Trilogy. There is a logical evolution of Trilogy's small-team/Supercoder approach to actual application development, but it is now has strict boundaries and is focused toward targeted industries.

The new focus is further reinforced by the embedding of development resources, and most other management and operational resources, within specific industry divisions. The cognitive map (and associated DSA model) highlights an evolution for some software houses from a product-focused horizontal application and organizational development approach, toward a vertical industry-focused approach. Such an evolution may be relevant for software houses providing specialized enterprise-focused applications and is relevant for other case-studies in this project (Zaplet in particular also moved from a horizontal to a vertical orientation during 2001).

On the other hand, some software houses in this project have preferred to remain horizontally focused even with its enterprise applications. Kana and NetDynamics/iPlanet have continued mainly as horizontally focused operations, although even in these cases there is increasing evidence of at least some broad vertical categorization of aspects of its products and services. Trilogy believes there is still great opportunity to further lower sales and marketing costs, and continues to develop new applications to address that challenge in the context of targeted industries.

Extension Questions

A set of additional questions was put to the interviewees for this case study, covering dominant logic, shared vision, leverage points, and strategic marketing issues. These questions were asked during the initial interviews and followed-up with further validation and commentary in the follow-up interviews.

Extension Questions – Dominant Logic

The interviewees agreed that Joe Liemandt has represented dominant logic for driving Trilogy's vision since its inception in 1991. Franke (2002) believed that he contributed to dominant logic for application development methodology, particularly during the 1990s, but also right through to 2002. Franke (2002) also identified John Price as the dominant logic for the creation of the "Trilogy University" to train and further develop Superprogrammers, during the mid-1990s. Hyams (2002) noted that around 1996, Scott Snyder was also noted as a dominant logic behind Trilogy's software development philosophy and in formalizing the actual software development systems.

However, since Trilogy's restructuring in 2001, dominant logic has devolved from Joe Liemandt (who still drives overall company vision) out to managers in the Divisions. There may be distinct "sub-logics" present within each Division as they work out its own visions and product ideas for its specific industry roadmaps. Hyams (2002) indicated that he provided dominant logic during 2000-02 for the core engineering related to Trilogy's applications, but not for specific products.

Extension Questions – Shared Vision

From Trilogy's inception in 1991 through to 2000, the interviewees believed that a lot of people in Trilogy contributed quite strongly to creating and extending shared vision. No one was shy about letting others know what they thought in Trilogy. However, Joe Liemandt was

always working on collecting these thoughts and articulating the shared vision. According to Hyams (2002), in 1998 the Trilogy operations team articulated and presented the vision. The team included Joe Liemandt, Neroj Gupta, John Price, Tom Carter and Chris Porch.

Shared vision has changed since Trilogy's restructuring in 2001, in that while Liemandt still collects and articulates the vision, Divisional General Managers contribute distinctively to it. There is a case to be put that Divisional General Managers may actually articulate a shared vision for and within its own Divisions.

Hyams (2002) noted that during 2001-02 there were other contributors to shared vision within Trilogy, but not necessarily in the Divisions. These contributors included Pat Kelly, Jim Abeld, Debra Ingram, Tom Rowe and David Phillips. From 1991-97, Trilogy had minimal relationships with third-parties and, apart from direct customization of its applications for targeted customers; it had not developed a shared vision with third-parties. However, as Trilogy pursued a rapid growth strategy to attain critical mass against larger ERP competitors, it developed an indirect business unit.

Trilogy worked hard with selected partners to build an enterprise-wide COTS e-commerce suite and shared that vision with those partners. Trilogy persisted with the indirect business unit, but complexities in managing the development of a full COTS suite across third-parties combined with the dot.com bust saw Trilogy abandon the unit in 2000. There may be opportunities for shared vision with key customers within business divisions, and possibly with strategic vertical industry system integrators and some developers, as the industry-focused approach evolves. A significant evolution of shared vision under the industry-focused approach is that while Trilogy always customized its applications for key customers from 1991-2000, the communication and interaction of ideas and testing with key customers is now directly explicated in Trilogy's software development methodologies.

Extension Questions – Leverage Points

Six leverage points were identified during the interviews. These are outlined in chronological order to include the following milestones:

1 Moving to Austin (1991-92)
2 Officially moving to "selling-chain" approach (Spring 1996)
3 Converting development to Internet applications (1997)
4 Internet business spin-off strategies (late 1999-2000)
5 Rebranding to e-commerce applications (2000)
6 Transition to vertical industry focus (2001).

1 Moving to Austin (1991-92)

Liemandt's decision to move the fledgling company to Austin was attributed mainly to family reasons. A direct follow-up leverage point from moving to Austin was hiring David Franke, whose reputation for software development and networking with various industry organizations enabled Trilogy to successfully pitch its configuration software to at least two substantial computer systems companies that needed its type of application. Franke immediately set about incorporating software development standards and completed a rewrite of the SalesBuilder application into a formal Version 1 release. Success with acquiring Silicon Graphics and Hewlett-Packard flowed on to signing up several large enterprises – and continuous rapid growth for Trilogy through 1992-99. Trilogy also attracted high-quality software developers who built on Franke's early standards into a "best-in-class" software development organization.

2 Officially Moving to "Selling-Chain" Approach (Spring 1996)

Liemandt's original vision articulated sales and marketing activities (and costs) as a "selling-chain". Liemandt believed that sales and marketing costs could be cut by around 2% through automation of this chain. From 1991-96, Trilogy's main contribution to automating this chain was the development and enhancement of its configuration and pricing application, SalesBuilder. The official move to focusing on the "selling-chain" really meant that Trilogy was

declaring its intentions to offer full "end-to-end"/"single product" application suites for the "front-office"/Selling Chain part of an enterprise's operations.

Declaring to the public and Trilogy's customers a full "selling-chain" offering was a clear recognition that sooner or later, the larger ERP application providers would extend out of its "back-office" applications toward full "end-to-end" overall enterprise application suites – meaning direct competition with Trilogy.

Liemandt knew that taking such a position meant that Trilogy had to grow very quickly to be at a critical mass when he expected the ERP vendors (SAP, Peoplesoft, Oracle, and Baan) to extend into the front-office area. Liemandt seemed to expect these incursions to occur around 1998-2000. Trilogy "bulked-up" rapidly through 1996 to early 2000, growing its "front-office" competitive position. Nevertheless, the larger ERP companies, especially SAP and Oracle made strong inroads into the Front Office area from 2001 to 2003, and now represent very strong threats to almost all major front-office application providers. Trilogy lost competitive ground during 2000, but restructuring toward targeted vertical industries was an effective competitive response for 2001 to 2003.

3 Converting Development to Internet Applications (1997)

Trilogy realized that its applications needed to become Web-based and switched to development using Java and XML in 1997-99. Trilogy had to hire large numbers of new programmers and staff with Web development skills. Hiring of new staff to develop Internet-enabled applications and systems doubled Trilogy's employees from 1997 to 1999. Trilogy's acquisition of the new Internet skills and capabilities enabled the company to pursue new Internet-related applications and businesses (see in additional leverage points).

4 Internet Business Spin-off Strategy (Late 1999-2000)

Details on Trilogy's Internet spin-off businesses can be found in Mandel & Austin's (2000) "Trilogy (B)" Case study note. However, the case study note was not included for detailed analysis for this project (that is, for DSA modeling and cognitive mapping) as it contained no reference to application development. One of Trilogy's founders, Christy Jones, had set up a "dot.com" venture, PcOrder.com, in 1995 using Trilogy's configuration software. As the "dot.com" boom reached its peak in 1999, Trilogy launched several ventures leveraging its "selling-chain" applications. Ventures included the following firms:

- CollegeHire.com
- CarOrder.com
- ApplianceOrder.com
- IveBeenGood.com
- InsuranceOrder.com.

Most of these ventures were launched toward the end of the "dot.com" boom and were closed during the bust in these businesses in 2000. Until 2000, Trilogy was a company completely focused on rapid growth and taking risks to support rapid growth. The "dot.com" bust and general downturn in technology stocks forced Trilogy to consolidate, restructure and eventually to move to a targeted vertical industry-focused organization.

5 Rebranding to e-Commerce Applications (2000)

Most ERP companies were slow to convert its applications to be Web-based, mainly because they were pre-occupied with implementing large and complex systems from around 1995 to 1999, in time to deal with Y2K contingencies. Most "front-office" application developers moved to Web-based applications from 1997 onwards, gaining about a three-year advantage over their potential ERP competitors. By 2000, several front-Office application developers had rebranded its applications as "e-commerce", meaning Internet-based applications.

Trilogy rebranded its applications as "e-commerce" solutions in 2000 joining this particular trend. Trilogy's commitment to the development and deployment of e-commerce solutions was deepened through joint ventures with large enterprises such as Ford (see Mandel

& Austin 2000). Some of these ventures and other close partnerships with large customers would contribute to the identification and establishment of targeted vertical industry business divisions in 2001.

6 Transition to Vertical Industry Focus (2001)

The dot.com bust of 2000, plus severe downturns in technology stocks, severely impacted on Trilogy's revenues, customer base and growth philosophies. Trilogy's response was to transform the company into five distinct business divisions, each developing and leveraging core Trilogy applications directly for enterprise customers in targeted vertical industry markets.

Development cycles and philosophies changed. Recruitment focused on experience in targeted industries and not so much on computing or IT knowledge or experience. At the time of interviewing for the case study (February-May 2002) it was too early to assess any outcomes from Trilogy's transition to a vertical industry-focused organization. However, by mid-2003 Trilogy had gained widespread recognition for its vertical solutions, from both its customers and several business and IT analysts. Trilogy added an additional dimension to its industry-based product offering, through linking its ongoing support revenues directly to its customers' own ROI measurements from using Trilogy's applications within its operations (see Kirsner 2002).

The addition of a specific customer ROI measurement to the development and deployment of an application suite can be traced logically back to Trilogy application development methodology and its Industry Roadmaps – but it is (at least in mid-2003) an apparently unique contribution to the software industry from Trilogy – and while it is beyond the scope of this project – it is a development that analysts and researchers are watching with great interest.

Strategic Marketing Issues

Three strategic marketing issues were identified from the interviews:

1 Vertical industry-focused marketing
2 Broadening product lines/application suites
3 Product management and product marketing roles.

1 Vertical Industry-focused Marketing

Trilogy's transition to a vertical industry-focused organization has exposed the organization to a different type of marketing approach, infrastructure and set of operations. For an organization that was deeply grounded in product development for horizontal use, understanding all aspects of an industry-marketing approach has been a steep learning curve. Trilogy has expressed its understanding of "industry marketing" by setting up industry-focused Business Divisions, each resourced with industry specialists in all management and operational roles. Perhaps Trilogy's philosophy of excellence in software development is now translated into establishing excellence in industry-based software development – including the supporting organizational structure to go with it. Hyams (2002) noted that shifting toward an industry-focused approach was not just about the organizational structure, but also moving Trilogy's brand to be seem as associated with targeted industries.

2 Broadening Product Lines/Application Suites

Both interviewees noted that during 1995-96, broadening Trilogy's applications out to a full "front-office" or "selling chain" solution represented a significant strategic marketing issue. Definition of what actually constituted the "front-office", and then developing the expanded application suites were major marketing decisions at that time.

Positioning Trilogy's expanded "front-office" suite to effectively compete with and/or complement the large ERP application suites was and remains a major marketing challenge for Trilogy. The strategic marketing issue of broadening product lines was not just pertinent for competition against large ERP providers. It was important for defining e-Commerce application suites as new enterprise e-Commerce applications emerged in the late 1990s and in 2000-02. Other software providers such as Kana branded its application suites as e-CRM, for example. Rebranding of "front-office" applications to be "e" applications took up a lot of strategic

marketing attention from 1998-2000. The larger ERP providers rebranded their offering as enterprise e-commerce application suites from 2000-02.

3 Product Management and Product Marketing Roles

The roles of Product Management and Product Marketing operating within a vertically focused organization were raised as significant strategic marketing issues. In almost all of the software houses researched for this project, Product Managers worked across both the development group and directly with customers and some key ISVs. As most of the software houses were start-up companies, product marketing was not usually a clearly defined group, apart from a marketing director and perhaps some marketing communications staff. Most software houses had defined sales groups that also worked with the product managers.

The main exception in the project is Intuit QuickBooks with an established marketing organization and highly defined product management and product marketing roles. Intuit QuickBooks was set up as a B2C or consumer marketing organization with these roles defined back in the 1980s – and it looks similar to software houses set up during that time (for example, Adobe and Microsoft). Trilogy's shift to an industry focused structure is similar to other B2B or business-to-business computer hardware and software companies who have attempted to move in a similar direction. At various times during the 1980s and 1990s and in the current decade, IBM, DEC/Compaq, Hewlett-Packard, SAP, TIBCO, and to a lesser extent the other ERP software vendor have all experimented it or established specific industry-focused business units or divisions.

There are variations on how much actual product development or customization is undertaken by industry business units or divisions. Some units may just undertake marketing and rebranding of the application for a specific industry with virtually no actual product modification. At the other extreme of a continuum, the core application may be quite basic and developed substantially within the industry business unit. A common variation may be that the core application is significantly customized by third-party partners such as SIs or industry-focused ISVs. In this variation, product management in the industry business unit or division is likely to have a significant coordination role in bringing together the resources for further application development and customization.

At the time of the interviews for this project, there were product management and product marketing for core applications development within each of the business divisions. Trilogy was trying to work out how both roles could work effectively with development groups to produce better industry-based solutions.

Summary

Decision-making issues associated with software development at Trilogy from its inception in 1991 to 2002 were captured and mapped in a package of DSA models, events chronology maps and cognitive maps and extension questions. The Trilogy case study provides deep insight not just into software development and delivery, but also in the underlying philosophies behind developing and changing actual software development over time. The package also highlights a transformation for Trilogy from a horizontal product-focused organization through the 1990s to a highly focused vertical industry organization in 2001-02. The interview process enabled extension of the case study timeframe to analyze Trilogy's transition in 2001-02, in substantial detail..

Key insights from the case study include the following observations:

- Transformation from product-focused to vertical industry-focused decision-making.
- Underlying software development philosophies and their contribution to decision-making
- Decision-making associated with anticipatory competitive actions
- Defining and redefining product lines/application suites.

Transforming from Product- to Vertical Industry-Focused Decision-Making

The package of models and maps provides deep insights into a transformation in Trilogy's decision-making processes from the late 1990s through to 2002. Between 1991 and 2000, Trilogy's business was based on very strong basic philosophies for software development, plus a clear vision to provide applications that would enable enterprises to reduce their sales and marketing costs. Trilogy's decision-making was occupied with rapidly developing applications to address an increasing range of "selling-chain" or "front-Office" opportunities.

The original vision presented huge potential revenues from large corporate customers – but only if Trilogy could grow quickly enough to have the resources and infrastructure to service these accounts. Trilogy managed rapid growth through a very strong internal focus on extraordinary productivity from its development teams. By 1996 Trilogy had expanded its applications into a substantial "front-Office" application suite.

Trilogy used several ventures to leverage its applications into new businesses. However, a number of these ventures were "dot.com" businesses launched at the peak of the Internet boom in 1999. While they logically leveraged Trilogy's front-office applications capabilities online, they were also early casualties in the dot.com bust in 2000. Trilogy's transformation to a vertically oriented organization was a strong response to the serious downturn in its revenues in 2000. The main links between the old horizontal Trilogy and the new vertical organization were key large customers who formed the basis for identification and establishment of Trilogy Business Divisions.

Almost everything about the Trilogy organization was reworked into the new Divisional structure. Trilogy's software development methodology was formally reworked to operate within each Business Division to create and deliver applications specifically tailored to Industry Roadmaps. Management and operations resources within Divisions were selected with emphasis on specific industry experience rather than IT industry experience. The DSA package for the Trilogy greatly benefited from multiple interviews which enabled the creation of new models and maps to effectively capture the transformation to a focus that was not necessarily an obvious evolution or extension from Austin's (1998) case study and Austin & Mandel's (2000) case supplement addressing Trilogy's Internet businesses.

Underlying Software Development Philosophies and Their Contribution to Decision-making

Trilogy set a vision to be the very best software development organization in the world. The organization's decision-making processes reflected and strengthened this vision. Underlying philosophies based on using small-teams of highly productive Superprogrammers, supported with automated testing routines and systems were factored into recruitment, training, and also to rapid application development and delivery. These philosophies were heavily tested but maintained as the organization brought in large numbers of Internet-skilled programmers and developers in 1997-98 in order to rapidly convert Trilogy's applications to be Internet-enabled and eventually to be branded as "e-commerce" applications.

Within 12 months of this influx of Internet-based resources, rapid software development enabled sufficient leverage of Trilogy's front-office applications into new online or dot.com ventures. Since 2001, most underlying software development philosophies remain, but are now incorporated in a software development methodology driven directly by specific vertical industry opportunities and requirements. There is less decision-making autonomy at the development team level, with substantial decision-making now residing at the Business Division Management Level with key inputs from various operations groups within that Division.

Decision-making Associated with Anticipatory Competitive Actions

Trilogy's decision-making processes highlight a strong concern for potential competitor actions. From 1993 onwards, Trilogy was tracking the main ERP vendors as potential competitors – and this was a major driving factor for pushing a rapid growth strategy through the mid-1990s. Trilogy's management was certain that there was small timeframe to grow the company to a critical mass from where it could withstand aggressive competition from the likes of SAP and Oracle and other large ERP vendors. Fear (and some actions resulting from it)

emanating from this perspective was identified and mapped in DSA analysis and cognitive mapping for this advanced case study. Decision-making associated with anticipatory competitive action was not highlighted in the transition to a vertically-focused organization. Rather, decision-making seems to have shifted toward directly addressing specific customer requirement, ahead of competitive action. That would not mean that focus on competitive action has gone, but it is now viewed more in an industry-based context.

Defining and Redefining Product Lines/Application Suites

Trilogy's founder, Joe Liemandt, articulated a strong vision for Trilogy at its inception – to develop applications that could cut sales and marketing costs by around 2% of revenues for a typical large corporation. Development of a full set of "front-office" applications caused Trilogy to realize the vision was not realistic for a start-up company. Trilogy commenced with an application for configuration and pricing of computer hardware and software. Nevertheless, within three years of the launch of the official Version 1 of its SalesBuilder application, Trilogy was able to offer a comprehensive "front-office" application suite and reposition the company as an "end-to-end front-office" vendor.

In 1997-98 Trilogy converted its application suite to be Internet-based and, by 1999-2000, rebranded its application suite as an enterprise "e-commerce" suite. Since 2001, Trilogy's applications have been defined, developed and mapped against Business Division specific industry and product roadmaps. The Business Divisions now work directly with key or "chartered" customers on large complex front-office solutions. These solutions may connect to larger ERP offerings or to specific systems developed by customers for specialized business functions (examples include: Automotive Product Management application suite for Ford; several pricing and contract management systems for various computer companies, a new global contract management and pricing system for British Airways; and a Channel Management System for the Prudential Insurance Company of America – for more details on these examples see Trilogy 2003a and b).

Summary – Additional Comments

The Trilogy case study highlights the decision-making associated with an organizational culture that exalts excellence in application development. DSA modeling augmented with events chronology and cognitive maps provide analysis and insights into rapidly building an enterprise application software house – and then transforming the whole organization toward a vertical industry-based focus.

The Trilogy case study explores the decision-making associated with combining a philosophy of "best-practice" application development with risk-taking in order to grow a company rapidly to prepare for an eventual and imposing competitive showdown with larger and more established corporations. The advanced case study also provides insights into decision-making from a big vision (automating enterprise sales & marketing activities) into a focused application offering (pricing configuration software) and onto a major provider of vertically focused (industry-focused) applications. The Trilogy case study offers some special contributions to the overall research project through analysis of explicit philosophies on application development that appear to be as fundamental to the organization as growing a strong and profitable software powerhouse.

Trilogy's strict and high-quality application development standards make the firm a highly internally focused software house. These philosophies and standards provide a special contrast to all the other software houses in the research project (with the exception of the early stages of the Intuit QuickBooks case study). Interviews for the Trilogy case study captured substantial insights beyond Austin's (1998) and Mandel & Austin's (2000) case study, particularly on Trilogy's new industry-focused application development cycles. Additional material provided by Trilogy enabled the most comprehensive insights into actual application development cycles out of all the cases for this project.

The creation and validation of a set of DSA models plus cognitive mapping outputs represent an effective mapping of strategic sensemaking over time and in multiple contexts. This extended DSA approach offers insights into turning an idea into an application in

conjunction with turning a small start-up business into a substantial and competitive enterprise software provider. Analysis indicates that Trilogy's executive management team was driven to take substantial business and application risks in order to quickly reach a "critical mass" such that they could take on severe competition from much larger and more established ERP software houses. Both Trilogy and its "selling chain" competitors perceived that its niche would develop into in a larger market for enterprise-wide application suites.

DSA models and selected maps for the Trilogy case provide unusual insights into a very strong logic for setting innovative application development processes and cycles. In fact, Trilogy branded its Fast Cycle Time (FCT) application development system.

Trilogy has a special system of recruitment and training to instill shared vision and empowered decision-making at the individual developer level in projects. The Trilogy case study offers mapping over time of decision-making associated with transformation from a "best-practice" product-driven business and application development approach into a vertical industry-focused business. The Trilogy case study provides valuable insights into establishing "best-practice" application development groups. Trilogy's strong organizational culture around "super-programming skills, capabilities and testing excellence through recruitment and training are explored in the case study. The interview process for the Trilogy case study provides valuable insights into transforming a software house from a strong product development focus across to industry-focused business groups and application development.

Trilogy Advanced Case Study Endnote

Trilogy is viewed as an innovative enterprise e-commerce (also seen in the media as an e-business) application provider and, although beyond the scope of the current project, new initiatives such as linking customer ROI to its ongoing use of Trilogy's applications could set the stage for further significant redefinition of what constitutes an "e-business" application suite (see Kirsner 2002). Trilogy is also working on defining new measurements for customer satisfaction relating to enterprise e-commerce application suites incorporating business value metrics (see Sawhney 2003).

During 2003, Trilogy packaged its Fast Cycle Time (FCT) application development methodology. FCT Web-based tools and supporting services are presented in Exhibit 9 (see also Trilogy 2003c).

Exhibit 9 Trilogy – "Fast Cycle Time" (FCT) Methodology Tools

From methodology to reality

Other project methodologies rely almost entirely on abstract processes and organizational discipline. The Trilogy Fast Cycle Time methodology, however, includes a set of leading- edge, Web-based "vision management" tools for automating critical aspects of the methodology, along with managed hosting services that provide end-to-end quality and availability for FCT projects:

Leadership.com is a multimedia Web-based application that allows executives to record project vision in a central location and makes the vision available to all project members.

eFeedback is a Web-based tool to elicit and consolidate user feedback, both on a Leadership.com site and on the target e-business Web site.

ePrioritize and eRankIt tools support project-level triage of features and requirements. The innovative combination of philosophy and technology that underlies the Trilogy Fast Cycle Time methodology keeps both project and vision synchronized, providing a sound foundation for e-business project management.

Managed hosting services fully integrate the FCT methodology with computing, storage, and networking resources, providing a solid foundation of quality, availability, and administrative responsiveness required for FCT project success.

Maximize business value with FCT

The Trilogy Fast Cycle Time methodology offers a superior approach to managing and developing e-business solutions, one that maximizes business value by constantly measuring progress and incorporating stakeholder feedback. To learn more about the FCT methodology, call the Trilogy Project Office at **1 .877.292.3266** or send email to **PMO@trilogy.com.**

Source: Trilogy (2003c)

Packaging FCT for Trilogy represents the translation of fundamental application development philosophy and decision-making directly into a set of application management services – an unusual but very interesting offering over and above enterprise e-commerce application suites.

Key Strategic Insights for Management Applications

The Trilogy case study provides special insights into a company with a very strong software application development culture – and it's efforts to change to become a more market focused business. Recommendations for management consideration and application include the following actions.

Managers should seek out, or develop, highly skilled software programmers. "Superprogrammers", when effectively deployed, can dramatically speed up commercialization and time-to-market for software applications. However, the Trilogy case also shows that deployment of superprogrammers without appropriate strategic focus is not necessarily a formula for business success either. Some form of directed market or customer focus is required to facilitate such supercharged development teams.

Sometimes there are opportunities to turn exemplary development processes into products or consulting services. Trilogy turned its fast cycle time (FCT) methods into a product and consulting service. Where a company has such strong product or application or service development capabilities, managers should explore scenarios for turning them into effective products and services.

Explore scenarios where the big players will counterattack. Smart companies growing quickly in a larger company's perceived space will eventually be attacked. Managers should explore ways to either make their product, service or technology so disruptive that the larger incumbent cannot meet the challenge, or to grow quickly enough to able to continue to beat the larger player in niches that suit the start-up.

When can ongoing proprietary application development keep winning? Is a full proprietary development approach actually now sustainable? Can organizations afford to wait to build capabilities in-house, or should they work more with third-parties to gain capabilities faster? Managers need to explore present and potential mixes of proprietary and third-party development for their application and service offerings.

References

Arnold, S.J. and Fischer, E., (1994), Hermeneutics and consumer research. *J. Cons. Res.* Vol. 21 (June) pp. 55-70.

Austin, Robert D. (1998), "Trilogy (A)", Harvard Business School, Boston, 9-699-034.

Capon, N., and Hulbert, J. (1975), Decision systems analysis in industrial marketing. *Indus. Mark. Man.* Vol. 4 No.2, pp.143-160.

Chandra, Mala (2001), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Redwood Shores, CA (13 December).

Christensen, Clayton M., and Michael E. Raynor (2003), *The Innovator's Solution: Creating And Sustaining Successful Growth*, Boston: Harvard Business School Press.

Franke, David (2002a), Personal Interview (Face-to-Face) with Hugh Pattinson, Austin Texas (13 February).

Franke, David (2002b), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Austin Texas (17 April).

Hulbert, J., J. U. Farley, J. A. Howard. (1972), Information processing and decision making in marketing organizations. *J. Mark. Res.* Vol.9 No.1, pp. 75-77.

Howard, J. A., Hulbert, J. and Farley, J. U. (1975), Organizational analysis and information-systems design: A decision-process perspective. *J. Bus. Res.*, Vol.3 No.2, pp.133-48.

Howard, J. A., and Morgenroth, W. M..(1968), Information processing model of executive decision. *Management Science*,. Vol.14 No.1, pp. 416-28.

Huff, A. S. (1990), *Mapping Strategic Thought*, Wiley, Chichester, UK.

Huff, A. S., and Huff, J.O. (2000), *When firms change direction.* Oxford University Press, New York.

Hulbert, J. (2003), Organizational analysis and information system design: A road revisited. *J. Bus. & Indus. Mark.*, Vol.18 No.6/7, pp.509-513.

Hyams, Chris (2002), Personal Interview (Telephone) with Hugh Pattinson, Sydney/Austin Texas (3 May).

Kirsner, Scott (2002), "How Do You Spell Value?" *Newsweek* (10 June) Vol. 139.

Langley, A., Mintzberg, H. Pitcher, P. Posada, E., and Saint-Macary, J. (1995), Opening up decision making: The view from the black stool. *Organ. Sci.*, Vol.6 No.3, pp.260-279.

Mandel, Marc and Robert D. Austin (2000), "Trilogy (B)", Harvard Business School, Boston, 9-600-123.

Sawhney, Mohanbir (2003), "How to Keep Your Customers Satisfied ; To learn if end users are happy, you have to ask the right people- and the right questions.," *CIO Magazine*, 16 (14), 1.

Thompson, C., Pollio, H.R., and Locander, W. B. (1994), The spoken and the unspoken: A hermeneutic approach to understanding the cultural viewpoints that underlie consumers' expressed meanings. *J. Cons. Res.* Vol.21 No.4, pp. 432-452.

Thompson, C. J. 1997. Interpreting consumers: A hermeneutical framework for deriving marketing insights for the texts of consumers' consumption stories. *J. Mark. Res.* Vol. 34 No.4, pp. 438-456.

Trilogy (2003a), Website, notes on testimonials and examples of Trilogy applications: Ford (http://www.trilogy.com/Sections/Extras/Downloads/Forrester.pdf); British Airways (http://www.trilogy.com/Sections/News/Press_Releases/PR_2003-05-12_0194.cfm); Prudential Insurance Company of America (http://www.trilogy.com/Sections/News/Press_Releases/PR_2000-010-10_0151.cfm), www.trilogy.com , all viewed 23 July.

Trilogy (2003b) Website, comments from Jack Welch (http://www.trilogy.com/Sections/News/Business_Press.cfm), plus extract from CIO (http://www.cio.com/archive/050103/netgains.html - linked to Trilogy Website) , and Newsweek. (http://www.trilogy.com/Sections/News/?NewsID=AR_2002-06-10_0188). www.trilogy.com , all viewed 23 July.

Trilogy Corporation (2003), "Introducing The Trilogy Fast Cycle Time Methodology: Delivering E-Business Value In Internet Time", http://www.trilogy.com/Sections/Extras/Downloads/FCT.pdf , Viewed on 28 November 2003

Wegner, D. M. (2002). The illusion of conscious will. Cambridge, MA: Bradford Books, MIT Press.

Weick, K. E. (1995), Sensemaking *in organizations. Sage,* Thousand Oaks, CA.

Whyte, W. F. (1943), *Street Corner Society.* Chicago University Press, Chicago.

Wilson, T.D. (2002). Strangers to ourselves: Discovering the adaptive unconscious. Cambridge, MA: Belknap Press of Harvard University Press.

Woodside, A. G. (2003), Middle-range theory construction of the dynamics of organizational marketing-buying behavior. *J. Bus. & Indus. Mark.* Vol.18 No. 4/5, pp.309-335.

Woodside, A.G. (2006), Overcoming the Illusion of Will and Self-Fabrication: Going Beyond Naïve Subjective Personal Introspection to an Unconscious/Conscious Theory of Behavior Explanation, *Psychology & Marketing*, 23 (3), 257-272.

Woodside, Arch G., Pattinson, H. H., and Miller, K. E. (2005), Advancing Hermeneutic Research for Interpreting Interfirm New Product Development, *Journal of Bus. & Indus. Mark.*, Vol. 20 No. 7, pp. 364-379.

Woodside, A.G. and Samuel, D.M. (1981), Observations of corporate procurement. *Indus. Mark. Man.*, Vol.10 No.2, pp.191-205.

Woodside, A.G. and Wilson, E. (2003), Case study research for theory-building. *J. Bus. & Indus. Mark.*, Vol.18 No.6/7, pp.493-508.