

# STRATEGIC PRODUCT DEVELOPMENT

A strategic approach to taking software products to market successfully

\*J van Zyl, \*\*A J Walker

*\*Rubico, South Africa*

*\*\*Software Engineering Applications Laboratory,  
Department of Electrical Engineering,  
University of the Witwatersrand, Johannesburg, South Africa*

**Summary:** *How can innovative quality software products be built on time and within budget? - a question frequently asked by management and product delivery specialists. Product features are known but competitors are hard on one's heels and the computer software industry is moving rapidly. Different, but integrated, standards are needed in the pursuit of building capable processes for delivering products. Project management principles can be applied, but not in isolation. Strategic Product Development is an approach that uses a number of industry standards in building an organisation capable of delivering commercially successful products. Strategic Product Development integrates the best of project management, product management, quality management, product positioning and marketing standards. The capability of processes within the approach are assessed and improved by using the ISO/IEC TR 15504 standard.*

## 1. Introduction

Typical questions asked throughout a product development life cycle include: Where to start? What exactly to build? How will concepts be integrated? Will the product perform? What are the dependencies and how will they be managed?

Product managers are faced with difficult, but challenging tasks when it comes to product innovation, concept development and product commercialisation. It is not always possible to listen to the customer and deliver exactly what is expected. The circumstances involved in obtaining the needs and wants for products are influenced by individual situations. Product innovation must take place in order to create products and services that potential customers do not yet know they need.

This paper presents an integrated approach that ties many concepts together in assisting the product development team to deliver world-

class software products. It also addresses indirectly the fundamental skills that are needed to make this possible.

Section 2 discusses the problems and solutions around product development.

Section 3 introduces two concepts that can assist with the product strategy development process.

Section 4 introduces the product development funnel. It shows how ideas and other strategic inputs are used to build products that can become commercially successful.

Section 5 discusses portfolio management as a technique to assist organisations in building the right products at the right time.

Section 6 shows how product lines are formed and managed. Project management principles are used to ensure project delivery.

Section 7 discusses some of the basic deliverables and work-products that should be



produced in software related projects. What does it mean to deliver a product to market?

Section 8 discusses what is needed once products are delivered to ensure long-term commercial success of the product. Some basic measurements are used to evolve product development organisations.

Section 9 brings a number of concepts together. The learning model is used as a medium to mature and evolve the product development business as a whole.

Finally some statistics are presented and Rubico's technology life cycle is discussed.

## 2. Problems and solutions to product development

Many books have been written, discussions held and debates organised concerning the reasons products fail. The public has been conditioned to accept that software projects are often associated with high business risks.

Discussing problems and failures, as discussed in this document, will only skim the surface of the vast amount of material available on this topic. It is not intended to be comprehensive but qualitative. These areas were selected in the light of product-based management.

What does it really mean when we say that a product or project failed? A definition might be: When a product, after being defined and committed to, did not give sufficient or any return on investment, or was terminated prematurely, then it is considered a failed product.

This definition obviously relates strongly to the commercial world, where capital (financial input) is needed for future developments.

Does this mean that all product and/or project failures are bad? Not at all - the learnings from failures are invaluable to an organisation, obviously assuming that learning took place and that there is a way to disseminate the knowledge gained from new experiences.

Failures can be classified into two categories [1] namely:

1. "Stupid, uncaring failures" in which the individual/s responsible need to be censured and
2. "Calculated risks or honest mistakes", where the individual/s learnt valuable lessons but took calculated risks.

Some problems and failures that have occurred in the computer industry over the last few years are discussed next. Some key concepts that are needed to solve these problems are presented in sections 2.2, 2.3 and 2.4.

### 2.1 Problems in context

The discussions in this section are not a comprehensive representation of the computer industry, but provide some background concerning the challenges facing product development.

In Glass [1], Kapur, president of the Center for Project Management, defined the seven deadly sins of project management:

- Mistaking half-baked ideas for viable projects.
- Dictating unrealistic project deadlines.
- Assigning underskilled project managers to high-complexity projects.
- Not ensuring solid business sponsorship.
- Failing to break projects into manageable entities.
- Failing to institute a robust project process architecture.
- Not establishing a comprehensive project portfolio to track the progress of ongoing projects.

Other issues include:

*Stock markets and venture capital are not sufficient to succeed:* Radical turnarounds are needed to pull companies back on track, for



example, Sapiens, Borland, Seer, Dynasty, Wang, IBM, 3Com, Oracle [1].

*Remember the basics:* Galileo International is a provider of electronic global distribution services for the travel industry. They ran a project to implement a new product called Agile for many months over its original estimated time frames. As management took control it became apparent that the basics of project management were overlooked. Fixes were introduced and the project delivered late without being cancelled [1].

*Not listening to customers:* Technical people are not skilled at listening to customers. They focus on the technological achievements and challenges and forget that customers need to use their inventions. Business processes need to be in place that will facilitate communication channels.

*Complexity causes confusion and costs:* Building overly complex products early in the product's commercial life cycle can cause failure. More highly skilled people are needed as products become more complex. Start small to test the market and then evolve the product.

*No definition of "complete":* Does a product ever reach a state of completeness? In the ever evolving world of improvement, few products reach a state of being fully complete. Market demand and potential influences the level of functionality that will be provided throughout the product development life cycle. Completeness is normally not defined before product development commences causing problems in coming to a close. "Deliver less more often", should be a slogan used throughout software delivery projects.

*Emotional attachments:* Teams become emotionally attached to their products. As time goes by and energy pours into development, individuals become tied to products and fail to make objective decisions. Sometimes losses need to be cut and direction changed.

## 2.2 Product management excellence

Product management is a strategic issue. Product development starts with ideas and the journey commences to commercialise these ideas into tangible products. Product management is defined to set the scene of what is meant by "excellence".

Product management can be defined as [8]: *"The product manager is responsible for a product, a product line, or several distinct products in a group. The areas the product manager has to deal with include packaging, branding, research and development, engineering, and production. The manager has to continually appraise the product's performance in terms of growth targets, market share, working capital targets, and return on assets managed"*.

In the light of this definition, the key characteristics required by product management include:

- Being responsible for a product: This requires total commitment and dedication to the product, and intrinsic accountability.
- Packaging and branding: How will the product be presented to the outside world? What makes people look at the features in lieu of satisfying their business problems?
- Research and development: Managing the functional and non-functional [9] attributes of the product. It might be that releases of the product are produced purely to increase performance or usability, for example, and not necessarily functionality.
- Engineering: Developing the product based on sound software engineering principles [5,10].
- Production: Developing the product to a robust commercial state.
- Appraising the product: Accepting and promoting continued feedback on the product. Providing sufficient feedback into research and development for improvements to take place.



Some management practices needed to complete the product manager's role are [11]:

- Planning and tracking of products and related processes.
- Achieving the right organisational structure based on current and future projects.
- Effective funding to get products to market.
- Building and communicating business cases to relevant stakeholders.
- Training and education of the teams.
- Organisational risk management.
- Operations management to ensure effective project integration and resource allocation.
- Customer and supplier interface management.
- Launching and institutionalising product lines.
- Technology forecasting and trend analysis.

Additional areas where product management must focus include:

- Direction setting and aligned of the organisation.
- The selection, training, and development of people to ensure a capable workforce whereby the vision can be realised.
- Through people, creating, shaping and influencing how work gets done.

Product stakeholders need to react when problems are identified in the development of products. It might be required that teams are re-aligned through horizontal movements, where team members are moved to other teams, or re-construction of the entire team, where the team is re-formed with new roles and responsibilities. Failure to act or react quickly to problematic situations can cause delays, possibly leading to product failures. It is better to make a decision, be wrong, and then fix the situation than to make no decision at all and let the world pass by.

Product management should ensure that releases are delivered more often in the early days of the product's life cycle. Great care should be taken that the architecture is sound and a platform is built for future development. This requires a focus on all product issues in the early days of delivery.

Some of the key skills required by strategic product direction setters include: being financially astute, having high level contacts, and technology skills.

## 2.3 Leadership roles

Leaders need to lay the essential product management foundations early on in the development cycle. To ensure proper execution, follow up reviews must be consistent and disciplined. There are "pulling" and "pushing" roles. Ideally leaders should spend most of their time by pulling (leading).

### Pulling roles:

- Direction setting: the vision, mission and business strategies need to be translated into a "picture" shared and understood by the entire team for effective execution.
- Product line architect: Building the right set of products is vital to keep the organization competitive.
- Portfolio manager: There must be a clear understanding of the products being developed and to be developed. The management of people, skills and knowledge is another portfolio that influences product success.
- Ultimate ownership lies with the leader. The decisions around which projects should continue and which not, become one of the most important aspects of a leader's role.

### Pushing role:

- Launching teams involve the approval of individual projects and version responsibilities in a product line.





- There must be a continuous source of *positive energy* from the leader. Skills and knowledge shortages must be identified openly and corrective plans put in place.
- Projects do not reach completion unless there is absolute commitment by the team and team members. Building commitment requires strong leadership and people capabilities.
- Finding and fixing problems that relate to process, people and technology can result in drastic improvement exercises. The leader needs to know when to change direction.

## 2.4 Management influence on product outcome

Organisational structures must be designed in such a way that the senior manager stays in close contact with the product development life cycle all the way up to pilot development. People such as Bill Gates, the Chairman and CEO of Microsoft, still become involved with the creation of ideas and then how the ideas must be developed.

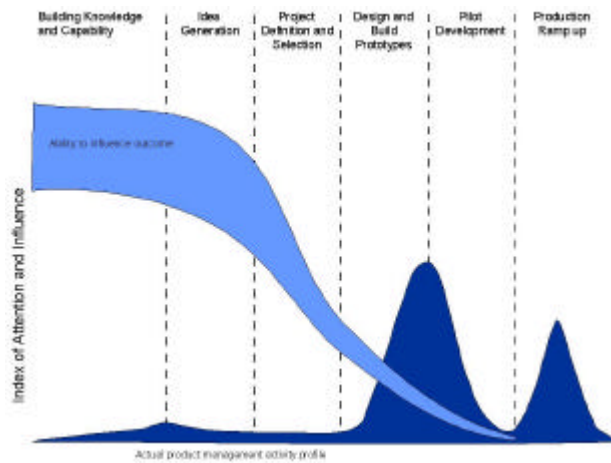


Figure 1 Management Influence in Product Development [2]

Management should become involved early in the process by building the right culture and capability to deliver innovative products. Clear direction needs to be provided all the way through the life cycle. The above figure shows that management normally moves into fire-

fighting mode as problems occur when money and time run out near the end of initial pilot projects. By then it is too late to influence the outcome significantly.

## 3. Concepts to assist with strategy

### 3.1 Building a *capable* organisation

The ISO/IEC TR 15504 [5] standard has a capability dimension that assesses the ability of processes to perform. Mature organisations use this as a method of benchmarking their capabilities against others in a certain industry. Each level has a brief reference to the individual and team abilities needed to perform. The five capability levels are:

Level 1: Performed: Processes are performed and people generally understand what needs to be done. Work products are produced.

Level 2: Managed: Processes produce work products and deliverables according to plan. Work products satisfy required quality and performance levels. Individuals working in this environment understand the implications of working with a defined process in assisting the delivery process. Teams are still unsure of how the final deliverables will be met.

Level 3: Established: Processes are performed based on sound software engineering principles. Processes are tailored to provide the best fit to what needs to be delivered. Defined process outcomes are achieved at this level. Individuals have obtained a clear understanding of the abilities needed to produce software products based on engineering principles. Teams operate within the limits of strategy and have a good understanding of what is needed to deliver predictably.

Level 4: Predictable: Defined processes are performed consistently to achieve goals and objectives. Predictability is high as the process is quantitatively managed. Individuals can deliver product after product predictably. They know when a process is out of control. Teams



can decide actively when to terminate a process under execution based on their capabilities.

Level 5: Optimising: The processes are repeatable and can be optimised to achieve business goals. Process innovation drives the ability of the organisation to become more competitive. The dimensions of people, technology, and process are all considered within the context of the organisation's purpose. Teams operate in the context of strategy and purpose and changes are dealt with proactively.

### 3.2 Building a “stage 4” business

The strategic role of teams in the business plays an important role in product development. Teams need to be balanced so that each product area can be a source of competitive advantage. Teams require the environment and direction to perform at their optimum. The risk is that teams can be pushed to perform at a level before they are ready [3]. An evolutionary approach is required where people and teams are educated and prepared over time to operate at level 5 (ISO/IEC 15504).

Weelwright and Clark [2] define four stages of evolution that a team goes through before achieving a strategic role:

Stage 1: Minimise negative potential, be internally neutral.

Stage 2: Match competitors, be externally neutral.

Stage 3: Back the strategy, be internally supportive.

Stage 4: Pursue a distinctive, sustainable advantage, be externally supportive.

Each stage is dependent on the previous, and is needed to facilitate essential learning. It is only once people are fully ready that they can operate purely within strategy as being the only direction-setting framework.

#### 3.2.1 Stage 1: Minimise Negative Potential - Be Internally Neutral

The team is put into a narrow operating role with little or no input into strategy. It is recognised that the team is necessary but is not essential. These people are not involved with anything outside of their environment and function inside of a tactical role.

#### 3.2.2 Stage 2: Match Competitors - Be Externally Neutral

The teams understand competitors and start to benchmark their performance. Outside assistance is still required although they operate at a higher level than stage 1. They still have a limited strategic role. The only way that management can allow teams to operate at this stage is when a *managed* level is reached.

#### 3.2.3 Stage 3: Back the Strategy - Be Internally Supportive

The teams can make decisions in the light of business strategy. They are actively supporting key business objectives. The function is directed and based on their abilities to satisfy the business objectives. People involved at this stage have input into the business strategy. Little outside influence is needed for this team to function. This stage will only be reached once the capabilities of the team are well understood, and they operate at an *established* level.

#### 3.2.4 Stage 4: Pursue a Distinctive, Sustainable Advantage - Be Externally Supportive

Customers and other stakeholders recognise that the team is a source of competitive advantage. The capabilities of the team differentiate the organisation from its competitors. Team leaders understand their strategic role and identify long-term trends and opportunities to build capability in anticipation of business strategy. The capabilities are replicated into other areas of the organisation. Delivering products consistently in line with



organisational strategy requires at least *predictable* processes.

In order to build a stage 4 product development organisation the following items should be considered:

- The capability of the organisation as a whole in order to achieve predictable outcomes. Strategic plans must be in place to make all within the organisation aware of the intention of the organisation to become world-class.
- A learning framework to assist in the accelerated growth of teams and individuals in teams. The culture of the organisation must dictate the fact that life-long learning is needed.
- People maturity and capability in dealing with the evolution as different teams can be at different stages of development. Clear development plans need to be in place to assist in the selection of people to perform roles.

#### 4. The product development funnel

Product conceptualisation and product strategic planning begins with strategic planning at Board level. The following diagram illustrates some of the stimuli that influence organisational strategy.

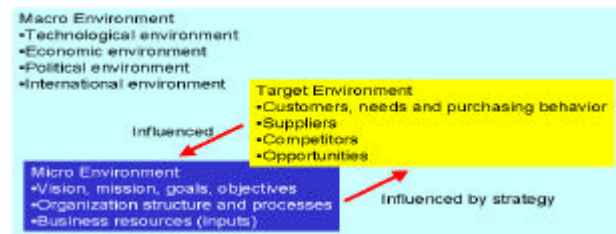


Figure 2 Business Environment

Organisations operate in an environment that influences how products are built and targeted. It is not within the scope of this paper to discuss the implications of the macro environment on product delivery.

Products and services are targeted based on a number dimensions (not a complete list):

- Targeted customers: What need does the product satisfy? How does the customer buy products of this kind?
- Suppliers: What are the influencing factors that will inhibit or enable product delivery?
- Competitors: Are the products positioned in such a way that there are clear differentiators?
- Opportunities: How can an organisation take advantage of the current environment in which it operates? New market opportunities and diversification can take place to generate opportunities.

The product development funnel is aimed at the micro environment. There are number of facets in the environment that must be considered to set direction and implement the funnel effectively [4].

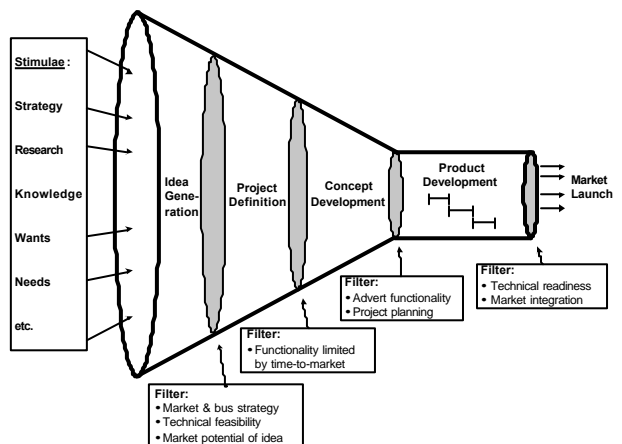


Figure 3 Product Development Funnel

Strategic planning and placement directly influence the product delivery cycle, as illustrated.

The funnel is used to take relevant inputs required to deliver products to market continuously. Inputs are accepted from:

- Existing customers: Inputs are provided based on current products and problems. These are normally the “needs” that make products work in the field.



- **Research:** Basic and advanced research projects are used to explore new concepts and generate innovations.
- **Market trends:** Trends are analysed to assess the ability of the organisation to capitalise on opportunities.
- **Strategy:** This is the interpretation and formulation of where the organisation needs to go based on the analysis of available information.

Concepts are assessed throughout the funnel to obtain a clear understanding of the commercial realities around taking the final product to market.

Filters are used to determine the inputs into the eventual product. Products are classified into product types. This is done to have optimal methods applied based on the type of product through the evolutionary and revolutionary product development process. Commercial impact is assessed on an ongoing basis. Predictability is important as the teams can draw from previous experiences.

All concepts and/or products under research must be channelled through a product development funnel. Organisational strategists are responsible for funnel integration and team formation in order to deliver products to market.

## 4.1 Product types

Products can be classified into four main categories. Products evolve over time and normally start with a breakthrough. These breakthroughs are used as platforms for other related developments. Platform products are used to build derivatives. Derivatives are supported over time up to the point of retirement.

Teams operating at stage 4 (section 2.4) can make decisions as to which product features are implemented when. They own the entire funnel and development that are required to commercialise the product.

### 4.1.1 Breakthrough products

Breakthrough products are products that are significantly different to existing product offerings. It normally means that the architecture has changed or components of an existing incomplete architecture were innovated. New value propositions can be presented to existing, but more likely, to new customers.

The life span is normally long and is seen to be a product that will be accepted only by early adopters. It could mean that the business is entering a new market.

### 4.1.2 Platform products

Platform products are built using breakthrough products. This is where the real customer value is realised. Customers will see that there is something innovative that can be used. New technologies and concepts are generally introduced with platform products.

Product families are formed to group the different technologies and concepts together. It could mean that the way the product is distributed needs to change.

### 4.1.3 Derivative products

Derivative products are built to leverage the investment in breakthrough and platform products. Extensions are added in an economical way but still within the quality limits. The product families become more stable as derivatives are created and deployed.

### 4.1.4 Support products

Support products are the small extensions that are needed to ensure product quality. They can also be used by the sales force to present additional features on products that are nearing the end of the life cycle.

## 4.2 Process standards

The ISO/IEC TR 15504 and ISO 12207 [5] standards can be used as a basis for processes in software related organisations. These must evolve with all facets [4] in the business to be





able to operate at a level where innovations can be commercialised consistently.

Processes must enable the business to move forward and not be a hindrance. Tailoring is required throughout the product delivery cycle to obtain effective results. One of the biggest risks is that management's understanding is not sufficient to lead their staff. Predictable process behaviour is required to deliver effectively.

Strategies should be in place whereby continuous cycles of education, experience acquisition and evaluations can be performed. Each cycle will represent a step towards becoming a world class software provider.

### 4.3 Project management standards

Having a good process does not guarantee product development success. The basics of project management must be in place to execute the process. Project management is used to deliver a certain set of deliverables within the product development funnel.

Some of the basic processes [6, 12] that must be in place in order to manage projects are:

- Scope management: Involves all aspects of scope definition and change management.
- Time management: Involves the planning of activities as defined in the process and includes the estimation of duration, schedule development and control.
- Cost management: Planning and control of project related costs.
- Quality management: Deals with the prevention, detection and tracking of issues.
- Human resource management: At project level staff planning handles all aspects of team selection and development.
- Communication management: All channels are defined and managed.

- Risk management: Risks are identified and classified. Risk management might involve performing research studies.

These processes will be controlled at product development funnel and delivery cycle levels. The macro product development plan shows all the product lines and interdependencies with regards to deliverables and resources. Product categories are kept together and innovations in the area of platform products are balanced.

### 4.4 Architecting the product line

Breakthrough products normally present significantly new and reworked architectural designs. It means that there are untested and unexplored areas that represent many risks, as the products that use the architecture mature. Architecture is the most vital area that needs stabilization early on in the product life cycle. Costly rework is minimized if there is a complete and stable architecture for a product family or product line.

Architectural snap-shots are needed to evaluate the appropriateness relating to product direction:

- Products available today.
- Products under development.
- Products of the future.

Considerations when working with product architecture include:

- Products of the future.
- Position: price/performance, market and competitors.
- Type: market need, contingencies.
- Timing: introduction, pace of change, market perception.
- Relationships: products for optimal integration and feature overlap.

Product architecture must be designed in such a way that software component and service-object reuse rates are high. This involves the tight integration of project deliverables



amongst product lines. The *Engineering process category*, as part of the ISO/IEC TR 15504 standard, has as part of its design process an architectural design base practice.

## 5. Portfolio management

The entire portfolio must be managed against business goals. Goals must include aspects that cover innovation in order to stay competitive. Radical ideas must not be excluded early on in the product funnel. These ideas can form the core of future products and present solutions, when implemented, to existing customer problems.

There are three macro-goals [13]:

- Value maximisation: there must be tangible value being generated by the portfolio. Some of the elements include: long term profitability, return on investment, the rate and likelihood of success and value created for customers that consider current and future potential.
- Correctly balanced portfolio: the organisation needs to define what balance means. Some of the elements that would be considered are: long-term vs. short-term projects, high risk vs. low risk, quick return, markets that the business operates in vs. future markets to compete in, different product types and different project types.
- Business strategic direction: the business is ultimately run based on its strategy that will include: financial and funding aspects, product direction, customer and target market, use of suppliers and partners and the tools and equipment that will be used.

### 5.1 Value maximisation

Expected Economic Value is a method used to maximise the commercial worth of the portfolio. It takes into account factors such as the:

- Expected commercial value of the product,
- probability of commercial success,

- probability of technical success,
- development costs of the projects that make up the product,
- commercialisation costs that include marketing and launching the product.
- and the nett present value of the product's future earnings.

It is not always feasible to follow this approach where up-front financial calculations determine a product's future. There must be times when management will continue with projects because of an intuition. Many organisations have built world class products even though they were seen initially as being a loser.

Organisations should build their own scoring models whereby the factors that are most important to them will be considered. A checklist might look like this:

- Expected economic value, including the assumptions that accompany the ratios and values.
- Core competencies in the organisation that can be exploited in such a way that innovations will occur that can yield major returns.
- Trends in the market can play a major role in deciding what to build and when to build it.
- Competitive advantage and market perception influences all decisions.

### 5.2 Balanced portfolio

It is not always possible to reach a balance amongst product lines and projects. Information might not be available or environmental conditions might influence decisions. Customers play a major role in decisions made about what gets built. Product management needs to make decisions that balance existing customer value vs. what is needed to generate new customers in the future.

Having a portfolio with breakthrough products only, for example, presents a major risk, as



there will be maturity and deployment problems. It is advisable to have a balance of different products at various stages of maturity in the product family.

A checklist with a weighting mechanism, defined by the product managers, might look like this:

- Will customers buy the product that is being developed?
- Sales method and financial impact.
- Technology issues that influence the satisfaction of existing customers.
- Will the product provide competitive advantage?
- Product type, maturity and the effort needed to make it commercially viable.

### 5.3 Strategic direction

Strategy in the software organisation has one major component that everything revolves around. Software is conceptualised, invented, designed and built by people. There is a degree of automation by using development tools, but the thinking part is human. By far the most expensive and strategically critical item is therefore the human component.

Strategy should consider the following items:

- Invention vs. delivery ability of people. Direction should dictate clearly how inventions are utilised and commercialised.
- A “top-down” strategic alignment means the vision and mission set strategy. Management allocates money.
- A “bottom-up” strategic alignment means that the company exploits its technology ability and shapes the business around it. When successful, the portfolio will look after itself.
- A “top-down, bottom-up” alignment means that the vision and mission sets the overall structure, but innovations could change that. Key people have input into the spending of money across portfolios. This

seems to be the best approach used by innovative and successful software companies.

## 6. Product delivery cycles

The product development funnel filters all the product concepts up to the point where development must start. This requires a clear definition of features that need to be developed over a certain time frame. Time-to-market, as defined by product strategists, is used to prioritise the product development cycle.

A product line is made up of a number of code lines. Code lines are for specific technologies and implement features for a specific release of a product.

Architectures are realised over a number of code and product lines. The architecture may even change over the life cycle of a product although it is not likely. A product is built once the architecture is defined and forms the basis of future releases.

Products are built over many releases. Each release adds functionality and becomes incrementally bigger and more complete. It does not mean that only one code line is used. Many technologies can make up one release of a product, and usage is based on relevant strengths and weaknesses.

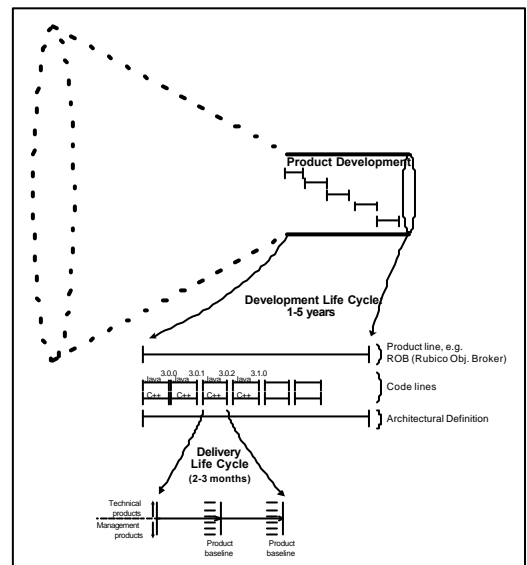


Figure 4 Product Delivery Cycle



Components that are constructed during the various code lines for a product line are used as software assets [11]. These assets include COTS (commercial off the shelf) software components and are all part of the final product family. Inter-product line dependencies are managed by having these well-defined components under configuration management early on in the funnel.

## 7. Evolutionary or revolutionary cycle

Innovative products are normally seen to be high risk when the time-to-market deadlines are aggressive. Think of the funnel as a *revolutionary product development life cycle* where evolution is challenged continuously. Challenges could involve the rethink of architecture resulting in an entirely new platform product. The key to having products accepted by customers is to have more releases early on in the life cycle with a tight feedback loop.

Specialists working on features in a release need to understand the importance of customer acceptance and feedback. The following diagram is aimed at having a part of or the entire product delivered.

Technical and management products are needed to establish a baseline. Baselines can be internally (technical reasons) or externally (marketing and business) defined. It is vitally important that the team understands the implications of bringing a product to a close. Forward and backward planning techniques are used to set the actual “work time”. For example, two days are needed to get the team ready and another three days to prepare the release.

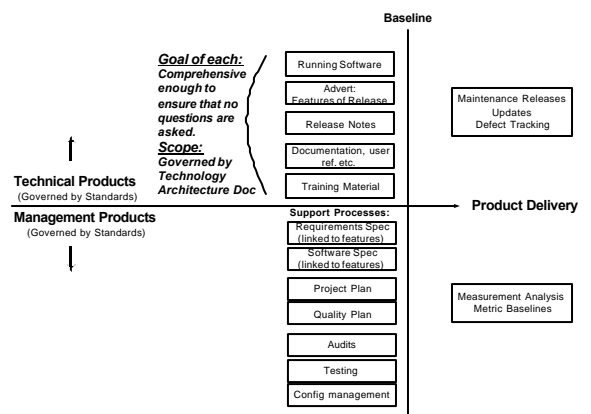


Figure 5 Working the Baseline

Management products are only used to assist in delivering the products the client sees (technical products are delivered to the client). Maintenance activities start once a product is baselined and released. All defects must be tracked to build a knowledge base about a product’s behaviour. Root cause analysis is critical in eliminating mistakes during the next build. Measurements are defined before the project starts. Metrics can be analysed and updated based on product, process and people behaviour. The learning model [4] is used to evaluate the findings.

Management reporting is done at the point of baseline. All measurement and metric elements are populated with the latest successes and failures. Time sheets are used to update the actual effort where project plans reflect the planning. The results of the project closure processes are fed back into the funnel and are used by product management to plan future cycles. A number of elements are considered before product concepts are explored in the funnel including: project closure feedback, customer issues, market trends and technology strategy.

## 8. Shipped product

Defects, enhancements and requests are managed once the product is shipped.

- Defects identified outside the development environment must be analysed in detail. The cause of the defect should be





investigated after which a corrective plan should be implemented.

- Enhancement requests can either be because the product does not have the selected features that the customer requires or the product is not used as intended by the author. The feature list of the product is used to manage the delivery of enhancements.
- Requests can be seen as wishes. What is required to make the product more useful?

Internally the team needs to close off the cycle of development before commencing with the project closure. It is at this time that all the measurement data are collated and metrics updated. Some product and product family related metrics include, and are classified into pre-release and post-release:

#### Pre-release:

- Product size and level of re-usability.
- Product defect types by phase.
- Re-work effort by phase.
- Project development effort.
- Process failures.
- Re-usable components by project and/or product.
- Productivity - time by month for activity group / item.

#### Post-release:

- Number of defects / enhancements / requests over time.
- Cycle Time - time to fix defects.

Statistical reports are produced weekly, monthly, quarterly and yearly by product managers. The results are evaluated before any improvement and/or education programmes start. Metrics are also updated at this time.

Processes to manage the evolution of standards and patterns in the various product areas are initiated after product shipment. This normally

coincides with the evaluation process in the learning model [4] as discussed in section 7. Processes are normally not clearly defined in the early stages of using new technologies and innovations (that is early in the product development funnel). There is interdependence between the kinds of products that are built and the kinds of processes used. Specialists and product managers evolve the processes with the products. Stage 4 organisations have the capability to understand this evolutionary process and manage the associated risks.

## 9. Learning the lessons

The Learning Model [4] is used as a tool to help the process of “*learning from one’s mistakes and successes*”. The basic principle of this model is “*Is there a better way of doing things?*” Understanding one’s own learning patterns will assist in increasing one’s knowledge and in return influence everyone in the organisation. This is one of the most important aspects of process improvement and maturity.

The learning model is implemented at various levels in an organisation or institution. The cycle of evolution is slower for direction setters than for the more focused and detail-oriented environments. Each area has its own patterns with which to deal.

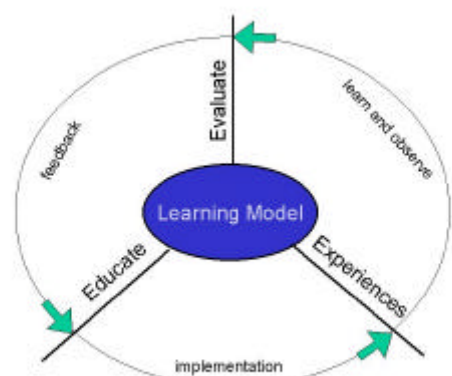


Figure 6. Learning Model Processes

The process normally starts with the recording of project experiences in ways that make future analysis possible. These experiences are used



during the evaluation process to determine where things can be improved without losing sight of the “bigger picture”. The evaluation process produces work-products that will be used to enhance, improve or re-engineer education or communication programmes. This facilitates re-alignment of processes, people and technology. At this point, new training candidates can benefit from the experiences, and existing staff can go through cycles of re-training. The re-training concentrates on new ideas and the areas that change.

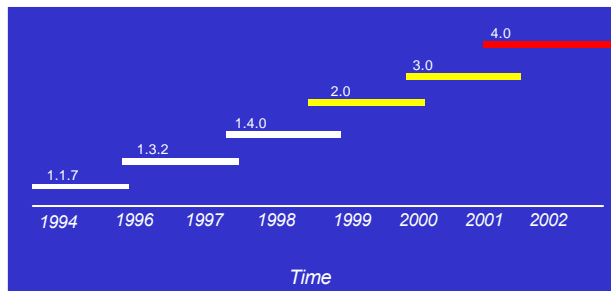
There is a reverse spiral that indicates the continuous feedback needed for the process to operate effectively. Experiences are based on what was taught during the education process. Evaluation is based on the experiences over a time frame.

Multilevel learning is when learning happens across two dimensions in the organisation namely: vertical (that is, from individual to group to corporate) and horizontal (that is, across product functional areas) [14]. The learning model facilitates both methods whereby different spirals integrate at different points in time.

Finally, the learning model is a high level process that encapsulates, or co-exists with many other models, architectures and frameworks.

## 10. Discussion

The concepts discussed in this paper have been applied at Rubico Technologies, the division at Rubico responsible for a number of innovative software products. Development of breakthrough products started in September 1994 and now, in July 1999, development is still continuing. In the time frame up to September 1994, the innovations that were used at Rubico were not viable due to certain technological and human limitations. The following diagram illustrates some of the major releases [7] of the Rubico Technology Product Family.



*Figure 7 Rubico Technology Product Family Life-Line*

Various innovations were produced up to delivery and many more are anticipated. Breakthrough products include the structure system, virtual database that is the evolved and extended structure system, rules engine that includes the calculation pad and state transition engine, generic masterfile, transaction engines, data cube and dynamic user interface definitions. Platform products were introduced at every major release. Today there are many projects running that are building derivative and support products.

With reference to figure 6, releases 2, 3 and 4 are very close together. Early innovations facilitated extended platform products that will evolve quickly as support products are built because more customers are coming on-line. Refer to Appendix A for a detailed view of events. It is anticipated that version 3 will involve various breakthroughs where version 4 will involve the volume deployment of the version 3 architecture. The assumptions are based on deliveries as measured and tracked over the last five years on the Rubico product lines.

Many software engineering principles had to be taught and implemented in the time frame between 1994 and 1997. Key commercial software development principles were new to the new recruits and created challenges for the education processes.

Delivery capability was (and is) challenged due to the sale of products to various customers that had to be implemented in the 1998 to 1999 time frame. Focus is firmly on the execution of early innovations.



Some of the fundamental findings in the evaluations that took place over the last six years were that:

- Academia does not prepare individuals properly for the “real world” of commercial software delivery.
- Decisions are made in circumstances of resource shortage, too little funding and unanticipated risks.
- Without a major investment in ongoing education the deliveries that took place would not have been possible. The assumption is that there will never be sufficiently skilled and knowledgeable people on projects.
- Strong leadership in the field of software engineering and product management based principles is essential for the delivery of commercial products.
- Strategy and long-term planning for overall organisational behaviour are essential to predict how the product life cycle will unfold. But can one really predict how technology will evolve?
- Technology specialists become tied into the innovation cycle without getting to deliver working products effectively. This could be because of the focus on working with tools and not developing tools.

Strategic product management concepts should always be implemented incisively with effective follow-up.

## 11. References

- [1] Glass R.L. (1999). Computing Calamities, Lessons learnt from products, projects, and companies that failed. Prentice-Hall. ISBN 0-13-082862-9.
- [2] Weelwright S.C. and Clark K.B. (1995). Leading Product Development, The Senior Manager's Guide to Creating and Shaping the Enterprise. Harvard Business School Press. ISBN 0-02-934465-4.
- [3] van Zyl J. (1998). People Maturity Model. Rubico and WITS University. (available on request)
- [4] van Zyl J. (1998). Software Engineering Applications Laboratory. WITS University.
- [5] ISO/IEC TR 15504. (1998). Software Process Improvement and Capability Determination, Part 5: An Assessment Model and Indicator Guidance.
- [6] PMI (Project Management Institute). (1996). PMBOK – Project Management Body of Knowledge.
- [7] van Zyl J. (1994-1999). Rubico feedback presentations to the engineering staff. (available on request)
- [8] Cary L. Cooper C.L. and Argyris C. (1998). Encyclopedia of Management. Blackwell Publishers. ISBN 0-631-20885-2. P528
- [9] ISO/IEC CD 9126. (1997). Quality characteristics and sub-characteristics.
- [10] ISO/IEC 12207. (1998). Information Technology Software Life Cycle Processes.
- [11] SEI. (1998). A Practice for Software Product Line Practice – Version 1.0, Software Engineering Institute & Carnegie Mellon University.
- [12] ISO 1006. (1997). Project Management Standard.
- [13] Cooper, Edgett and Kleinschmidt,. (1998). Portfolio Management for New Products. Perseus Books. ISBN 0-201-32814-3.
- [14] Weelwright S.C. and Clark K.B. (1995). The Development Challenge. A Harvard Business Review Book: HBS Press. ISBN 0-87584-609-2.



## 12. Author Contact Details

**Jay van Zyl** is the director responsible for technology, research and education processes at Rubico. He has a background in electronic engineering, information technology and internal auditing.

He is currently involved with the management and research of education programmes for business and technology professionals. These education programmes cover the entire software engineering spectrum and specific software development methodologies. He is also involved with the research and development of implementation, development and change management methodologies at Rubico. Being at the head of research: product direction, software requirements engineering and strategic planning for the Rubico software product suite, is also managed by Jay.

His research interests include: organisational process, design and behaviour in the context of software engineering and product delivery, process innovation and strategic product management, process assessment, process improvement, people measurement and business modelling methodologies.

Jay is a post-graduate, part-time student at the Software Engineering Applications Laboratory, Department of Electrical Engineering, University of the Witwatersrand, Johannesburg, South Africa.

He is a member of the ISO/IEC workgroups and is a participant in the SPICE (ISO 15504) Trials. He also serves on the board of advisers for the International Quality Assurance Institute (QAI) and is a Certified Quality Analyst (CQA).

*Contact details:* Jay van Zyl, Rubico, P.O. Box 5016, Rivonia, 2128, Republic of South Africa. Office Phone: +27-11-808-1000, Fax: +27-11-808-1101, Internet E-mail: [jay@rubico.com](mailto:jay@rubico.com)

Internet URL: [www.rubico.com/JayVanZyl](http://www.rubico.com/JayVanZyl)

**Alastair Walker** is presently an associate professor in the Department of Electrical Engineering, University of the Witwatersrand. He was responsible for establishing the Software Engineering Applications Laboratory in 1988. Since that time, the SEAL has become known as the leading provider of advanced professional courses on topics in Software Engineering, particularly in the areas of application of object-oriented design methodology in industrial applications and in software quality systems management. SEAL was the first academic institution in the world to receive an ISO 9001 certification for software development.

He is a Certified Quality Analyst, a Certified Software Quality Systems Auditor and is the country's only Certified Auditor in Education and Training.

He is presently a member of the Advisory committee and the Information Technology Committee of the South African Bureau of Standards. As a member of the Southern African Council for the Certification of Quality Systems Auditors, he advises on issues related to training of software quality system auditors.

He has recently been appointed convenor of the SABS work group involved in the development of the new international standard for software process assessment and for the upgrade of the 1991 version of ISO 9000-3 and is a South African representative on the ISO/IEC/JTC1/SC7 work group. He is the secretary of the Software Quality Specialist Division of the South African Society for Quality.

*Contact details:* Professor AJ Walker, Software Engineering Applications Laboratory, Department of Electrical Engineering, University of the Witwatersrand, Private Bag 3, PO Wits, 2050 Johannesburg. Phone: +27 11 716-5469, Fax: +27 11 403-1929, Internet E-mail: [walker@odie.ee.wits.ac.za](mailto:walker@odie.ee.wits.ac.za)





## Appendix A. – Summary view of *Rubico Technology Products* life line

Time frame	General happenings
1993	Pilot projects, anticipated breakthroughs based on past research. Education focus, preparing environment. Funnel started and concept development explored.
1994	Product architecture and initial infrastructure, breakthrough products. Education process, technology and architecture. Standards, patterns and process implementation focus. Funnel explored further and initial projects put in place to evaluate breakthroughs.
1995	Product innovation and concept development, breakthrough and platform products. Process and measurement focus, commercial product development principles introduced to team. Learning process evaluations took place to accelerate development and new product funnels kicked off.
1996 First commercial ready release 1.1.7	Product delivery and stabilization. Deployment and testing focus. Product commercialized at end of first development funnel. Sold one major account with three sites.
1997 External sales Release 1.3.2	Product innovation and new markets research, mostly in the areas of performance. Total integration focus, 3 <sup>rd</sup> party reporting tools, etc. Evaluations took place that resulted in major re-education of all staff. Sold two accounts and twenty sites with implementations of over a year each.
1998 Rubico Releases 1.4 and 2.0 (late 1998)	Product portfolio enhancement from breakthroughs – reporting tools ver 1 through 3, replication. Internationalization and benchmarking. Internal re-design phase 1. Sold seven major accounts with implementation times ranging twelve to twenty four months. One hundred and thirty three sites to be implemented.
1999 Rubico Release 3.0 – ramp-up from ver 2, Release 2 updates 2.1 etc.	Independent and synchronized product delivery. Logistics and deployment. Product usability and stabilization. Application partitioning. Sold two major accounts with thirteen sites. Ready for next wave of major account sales.
2000 Rubico Release 3.n – commercial release	International deployment and issue handling. Product architecture re-implementation moving towards ver 4. Dynamic application partitioning. Non-functional attribute focus i.e. performance, scalability, robustness, etc.
2001 – 2005	International volume deployment and logistics management – breakthroughs anticipated. Product stabilization and additional platform and derivative products.

