Appendix B

A Pattern Language for Strategic Product Management

This appendix introduces a pattern language for strategic product management (see Table B-1). What is unique about the patterns is that they allow the marketect and the tarchitect to bridge any gap between their respective disciplines.

TABLE B-1 Strategic Product Management Patterns

Problem	*Solution*	*Pattern*
How do you segment your market?	Create a visual representation of the market(s) you're targeting.	Market Map
What is the right frame of reference for time in your problem domain?	Identify the events and rhythms of your market/market segments.	Market Events/Market Rhythms
How do you ensure that the right features and benefits are being created for your target market(s)?	Create a map of the proposed features and their benefits. Tie these to the market(s) you're targeting.	Feature/Benefit Map
How do you manage the evolution of your technical architecture?	Create a roadmap of known technology trends. Include specific and well-known changes you want to make to your architecture so that everyone can plan for them.	Tarchitecture Roadmap

Applying The Patterns

Figure B-1 captures the relationship between the patterns as they are applied. The order shown is one that has worked for me. Like most diagrams that suggest an ordering, the figure fails to show the dynamic way these maps were built. In most cases, I recommend starting with a Market Map pattern, primarily because there is so much confusion among product development teams regarding the target market. However, if you're working in a mature market, or if your marketing communication department already has a precise calendar, you may find that starting with the market events and market rhythms patterns is the best way to begin.

The most important point is that instead of arguing which pattern should be first, you should simply pick one to get started. This is because the patterns are part of a

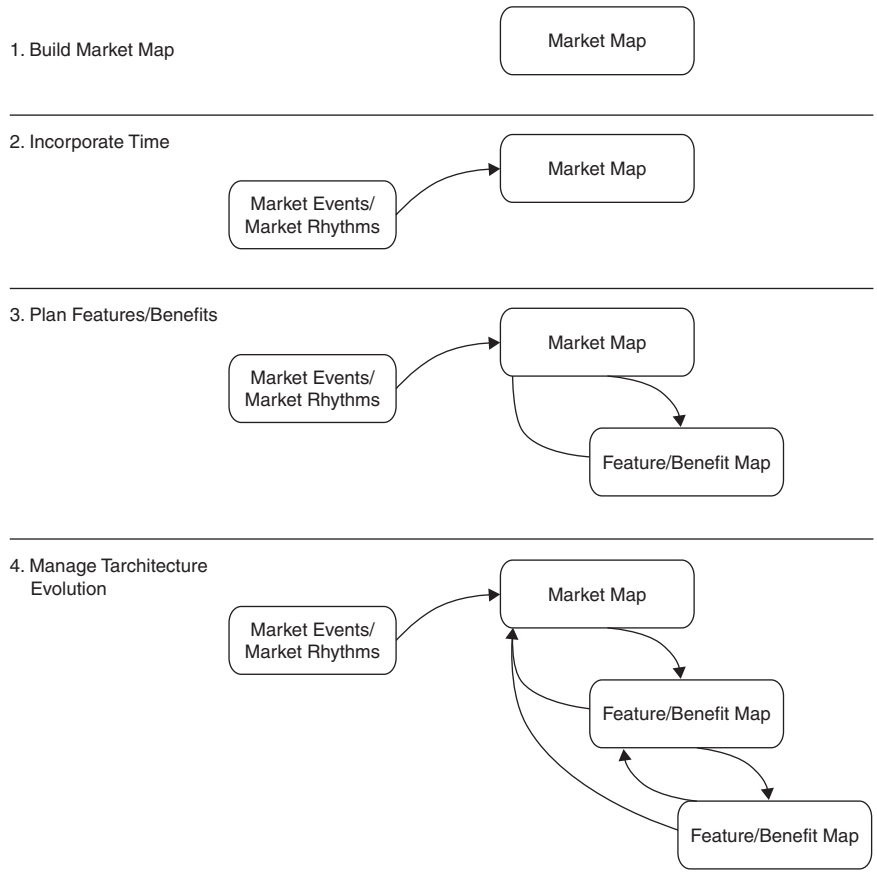

FIGURE B-1 @@@Author to provide caption here@@@

system. Like all complex systems, this one is characterized by feedback loops, shown in light gray in the figure. It is almost guaranteed that the application of any given pattern will slightly modify the earlier application of another one. Instead of focusing your efforts on making a "single" diagram correct, try starting by identifying one market segment—just about any will do—and then add some market events or features. Try to increase your comfort that the data you need to make sound decisions will emerge, provided you keep iterating over the results.

Discontinuous events, such as a competitor releasing a new version faster than expected, a new conference forming around a new industry, or the introduction or maturation of an exciting technology, are likely to motivate a series of coordinated changes in all of these diagrams. For example, in applications that rely on voice technology, the maturation of Voice XML or SALT that is first captured on the tarchitecture map may cause upward ripples in all of the other maps. You might be able to reach a new market with an awareness of these new technologies, or you might be able to provide some "killer feature" by adding new support for these standards.

Capturing the Result

I've found that the best way to capture the result of applying these patterns is through a series of large charts located in a publicly accessible dedicated space. A stylized, condensed example of such a display is shown in Figure B-2. The large question mark represents what happens when marketing identifies an unmet need requiring a technology or capability. The final row of the grid is the addition of the real schedule, or that schedule that has been communicated to customers and salespeople perhaps through

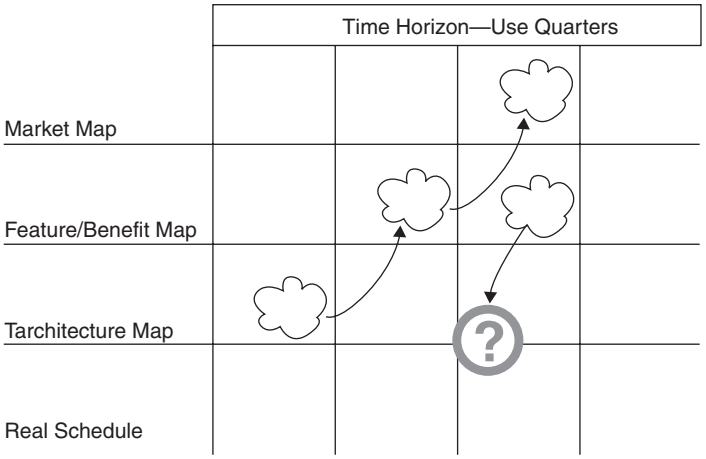

FIGURE B-2 @@@Author to provide caption here@@@

the product roadmap. Placing the real schedule along the bottom of the diagram ensures that everything is being considered pragmatically. Not shown in this example are market events/market rhythms. these should be added in your application of these patterns.

The Market Map Pattern

Context

You have an idea for a new product and you're trying to understand its potential. You have an existing product and you want to make certain that you're marketing it in the most effective manner possible.

Problem

How do you segment your market?

Forces

- Market segmentation is hard because
  - Existing markets change.
  - Predicting emerging markets is as much art as it is science (who knew lasers were for playing music?).
- Market segmentation is critically important because
  - If you don't segment your market you run the risk of trying to serve all markets, which is almost certain failure.
  - Different market segments require different solutions. You need to focus to win.
- You can't identify the most profitable segments if you don't segment well.
- You can't meet the needs of every market.
- Usability requires an understanding of the market you're trying to serve.

Solution

Segment the market by creating classes or groupings of users who share similar characteristics and/or attributes. The characteristics include critical needs, buying patterns, and various attributes that are important to you. The attributes in a consumer market might be age, household income, Internet connectivity, and technical literacy. In a business market they might be revenue, number of employees, geography, and so forth. Name the segment on a piece of paper and then write down its most important

descriptive characteristics. Large Post-It notes work well because they can be easily ordered.

Concentrate first on the actual users of your current product. (If the users are not the customers—people who have purchasing authority—you can address this at a later date.) In this process you will identify common "points of pain" or problems these users face. The results provide input to your Feature/Benefit Map. More important, though, is that you must solve your customers' problems.

When you begin this process try to make your segments as well-defined as possible. This will help you focus your efforts. As you examine each segment to make certain it is a viable (profitable) target, you may want to combine it with other segments. A fine-grained approach to market segmentation will give you more flexibility in combining segments should this be needed.

Once you have a reasonable number of segments (usually between 6 and 12) them in terms of which segment you will be addressing first relative to the actual and/or contemplated features of your product. Even before the product is finished some segments will naturally emerge as "easier" to address than others. This may be because of existing relationships (such as channel and/or customer relationships) or because it is simply easier to build a product that pleases a certain segment. As you complete the other maps in this pattern language you can adjust the timeframes associated with the target segments.

Provide the market map to all team members, especially user interface designers and QA. User interface designers need the map to understand the needs of the customer. QA requires it to make certain that they are organizing testing according to key customer priorities.

Resulting Context

Your market is segmented at a sufficient level to support strategic planning. As the needs of one segment are addressed, the next segment can be more precisely analyzed in preparation for the product cycle.

Related Patterns

- Market Events/Market Rhythms
- Feature/Benefits Map

Market Events/Market Rhythms Pattern

Context

You are trying to establish the market window for a given release *or* an ongoing series of releases. You have a Market Map pattern to help you explore specific market segments.

Problem

How do you choose a good target release date? How do you establish an ongoing cycle of releases?

Forces

- Customers usually can't accept a release at any time of year. For example, retailers won't accept new software releases during the holiday selling season.
- Customers can't absorb releases that occur too quickly, yet they get frustrated and concerned if releases happen too slowly.
- Developers hate arbitrary release dates.
- Sales activities require plenty of lead time.
- Releases that are too short are almost always bad—low quality or incomplete.
- Releases that are too long are almost always bad—too many features and lost revenue and market opportunities.
- Organizations (developers, support, QA, release manufacturing, and others) that go too long without releasing a product lose valuable skills.
- Organizations that go too long without releasing aren't fun places to work.

Solution

Identify and record the key events and rhythms that drive your market. Every domain has them. For example, Comdex and CEBIT are important international conferences that drive many end-consumer computing devices. Consider the following when you're searching for important events.

- Conferences (technical, user group, and so forth)
- Competitors' press releases
- Topics of special issues in publications that focus on your domain and relate to your product or service

Events that are held periodically, such as Comdex, also create and establish the rhythm of the market. If you're a successful consumer electronics vendor, Comdex and CEBIT form the foundation of a yearly rhythm known by all market participants and driving all company activities. Other examples of marketplace rhythms include

- The end-of-year holiday season
- In the United States, the school year
- In Europe, summer vacations

Once you have identified marketplace events and rhythms use them to create the timing and rhythm of ongoing releases. I've had good luck with regular release cycles

of between nine and twelve months. Broad software categories, such as high-end and enterprise systems, often have major releases every twelve months, with dot or maintenance releases following three to four months thereafter. Some software categories, such as operating systems, don't seem to have a rhythm.

The maturity of the market segment also affects the release cycle. Immature markets tend to have shorter release cycles and more events—a reflection of the learning going on among market participants. Mature markets tend to have longer release cycles and more established rhythms.

Resulting Context

Developers are happier because they know that marketing isn't making a date out of thin air. Commonly known dates have an energizing and engaging effect on the entire development organization. Customers are happier because they can engage in realistic strategic planning. They know that sometime in the third quarter they will be receiving a new release and can plan accordingly.

Related Patterns

- Market Map

The Feature/Benefit Map Pattern

Context

You want to make certain that key marketing objectives match key development efforts. You have a Market Map to identify the key market segments you're targeting.

Problem

What is the best way to capture compelling features and their benefits for each market segment?

Forces

- People often think they understand the key features and benefits for a given market segment when they really don't.
- A feature may be a benefit to more than one market segment.
- Features that apply to multiple market segments may not provide the same perceived benefits to each one.
- Developers tend to think of features (cool), while marketing people tend to think of benefits (compelling advantages, reasons to purchase). This gap often results in poorly designed products and/or products that can't win in the market.

- Developers need to understand the nature and intent of *future* benefits so that they can be certain the tarchitecture is designed to meet them.

Solution

For each market segment capture the key features and benefits that motivate customers to purchase the product. Display these on the market map It is crucial that you list features and benefits together. Omitting one gives the other inappropriate influence.

Choose an ordering for your map that makes the most sense. I've had good results ordering first by time and then by difficulty/complexity. Others get good results from ordering by what the market wants or what sales can sell. Paul Germeraad from Aurigin Systems, Inc., has organized features into a product tree where the edges of the tree are the features in the last release. The features of the next release are placed around those edges. Paul draws lines around the features proposed for the next release. One advantage of this visualization is that the entire company can feel good about the growth of their product. Another is very practical: The expanding size of the perimeter correlates with the growth of the product development organization. As the product tree grows, so do the needs of the team in caring for and feeding it (including the maintenance team).

Resulting Context

You have a representation of the key features and associated benefits needed to attack target market segments. This will provide the technical team with the data they need to update their tarchitecture map so they can realize these features.

Related Patterns

- Market Map
- Tarchitecture Roadmap

The Tarchitecture Roadmap Pattern

Context

You are building an application expected to have multiple, ongoing product releases. You have a Market Map and a Feature/Benefit Map to identify specific markets and the features/benefits that they want. You may have an existing architecture that supports one or more markets.

Problem

How do you manage/leverage technological change?

Forces

- No matter how well an application has been architected, changes in technology can invalidate prior assumptions.
- Technologies usually appear on the horizon with enough time to accommodate them if they're planned for.
- Developers like to understand where they are headed.
- Developers like to learn new things.
- Developers want a way to manage the tarchitectural evolution of poorly implemented features. The want a way to make both the poor feature known to others and register their desire to change it.
- Technology can enable new features that marketing may want.
- Marketing may demand features that can be supported only by adopting a new technology.
- Competitors' adoption of a new technology may put you in a disadvantageous, reactive state.
- Technical people will argue over emerging technologies. Sometimes the arguments are a way of learning more about the issues. Most of the time the only way to reach consensus is to give them plenty of time to discuss the issues.

Solution

Create a technology map that shows how your architecture will evolve. It should relate to the market map and feature/benefit map by showing how specific technologies produce benefits that are desired by key market segments.

Review this map whenever important milestones are realized and no less than once every six months. Examples of important internal milestones include code freeze, product shipment, and when 50% of current customers have upgraded to the most recent version. Examples of important external milestones can be a competitor issuing a new product release, new patent discoveries, whenever a member of the technical staff identifies a significant discontinuous technology, or whenever market events occur.

The creation and management of the tarchitecture roadmap requires that at least one member of the team be scanning the external environment for new developments in the field.

If marketing has identified a feature that cannot be supported by existing technologies (either directly or because of performance/cost curves) the tarchitecture roadmap can help the team maintain focus by periodically scanning the environment to see if any new technologies have emerged that meet this need.

Resulting Context

The good news is that you will identify promising technical futures for your product. The bad news is that unless your team has sufficient discipline they will want to explore every possible future—the dreaded "shiny new object" syndrome.

Related Patterns

- Market mAp
- Feature/Benefits Map