



Kuusi vuotta sprinttien pyörteissä - F-Securen kokemuksia ketteryyden kierteistä

Oliopäivät; 2009-12-09

Pirkka Palomäki, CTO

Agenda

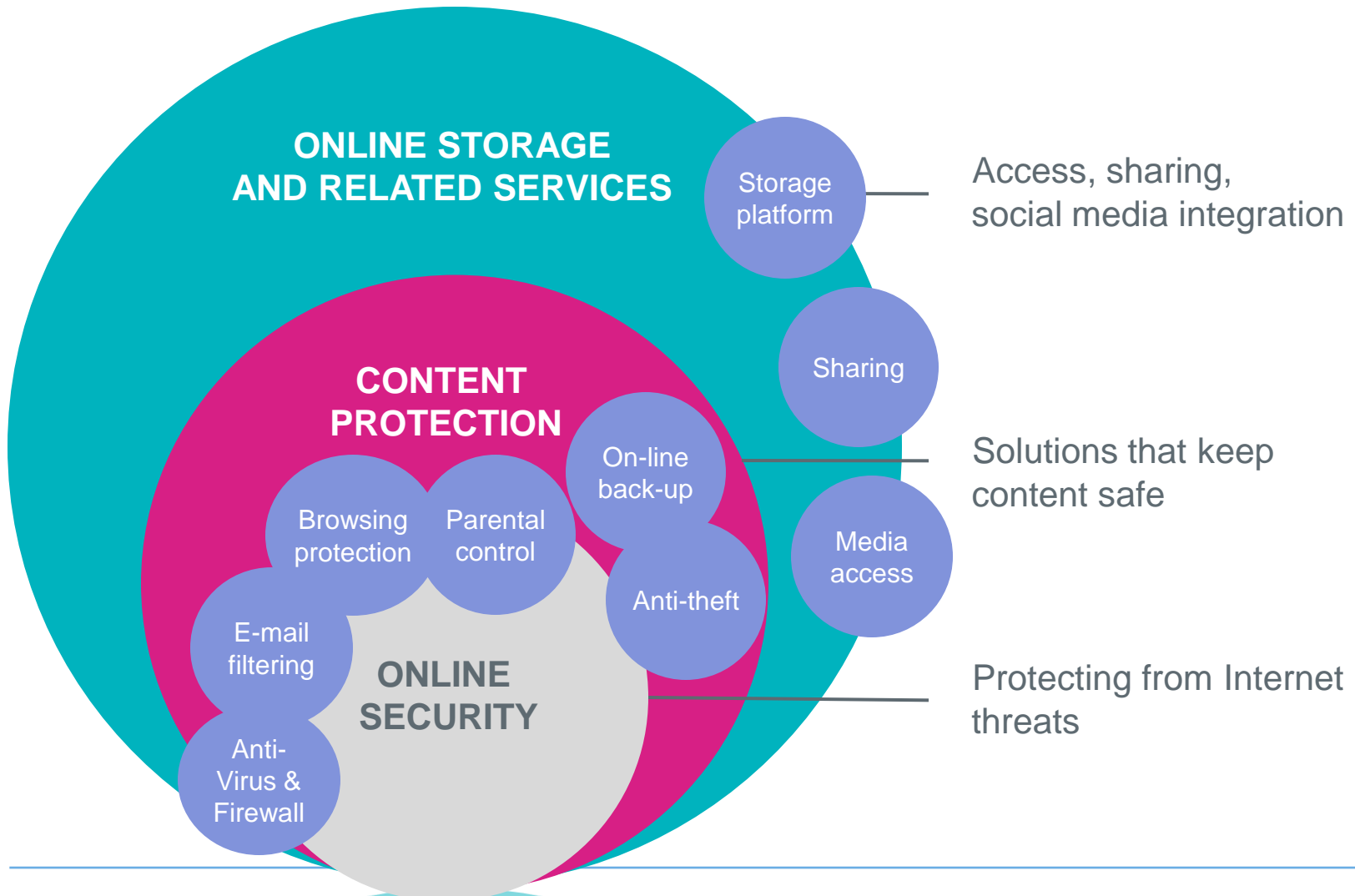
1. Case Company Background
2. Initial attempts to change the game
3. Swinging the pendulum
4. Initial results
5. Recent development
6. Summary

Agenda

1. Case Company Background
2. Initial attempts to change the game
3. Swinging the pendulum
4. Initial results
5. Recent development
6. Summary

Portfolio expansion

(PCs, mobile phones, new devices..)



An overview to the portfolio

Seeding and marketing applications:

- Health Check
- Online Scanner
- World map



Systems, Services & Tools:

- VAS delivery platform
- In-the-cloud & 24/7 security services
- Customization & integration services
- Update, delivery & storage infrastructure
- Service delivery automation
- Support tools and information systems

Value added services:

On-line backup, storage & sharing	Parental control	Anti-Malware	Anti-theft (lock , wipe & trace)
System performance tuneup	Vulnerability check & system updater	Browsing Protection & Privacy	Firewall, intrusion prevention & application control
		Email safety/productivity	

R&D @ F-Secure

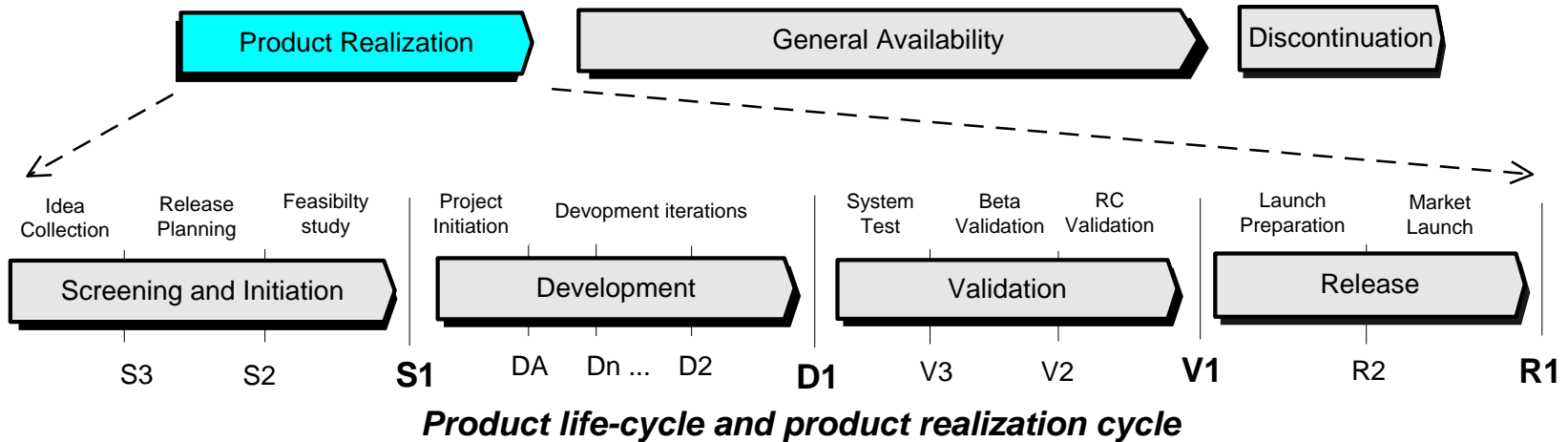


- 300+ people
- Located in 5 offices in four countries
- Typically around 20 concurrent projects / release efforts
- Many supported Operating Systems, OS version and > 20 language versions
- Common components & product platforms

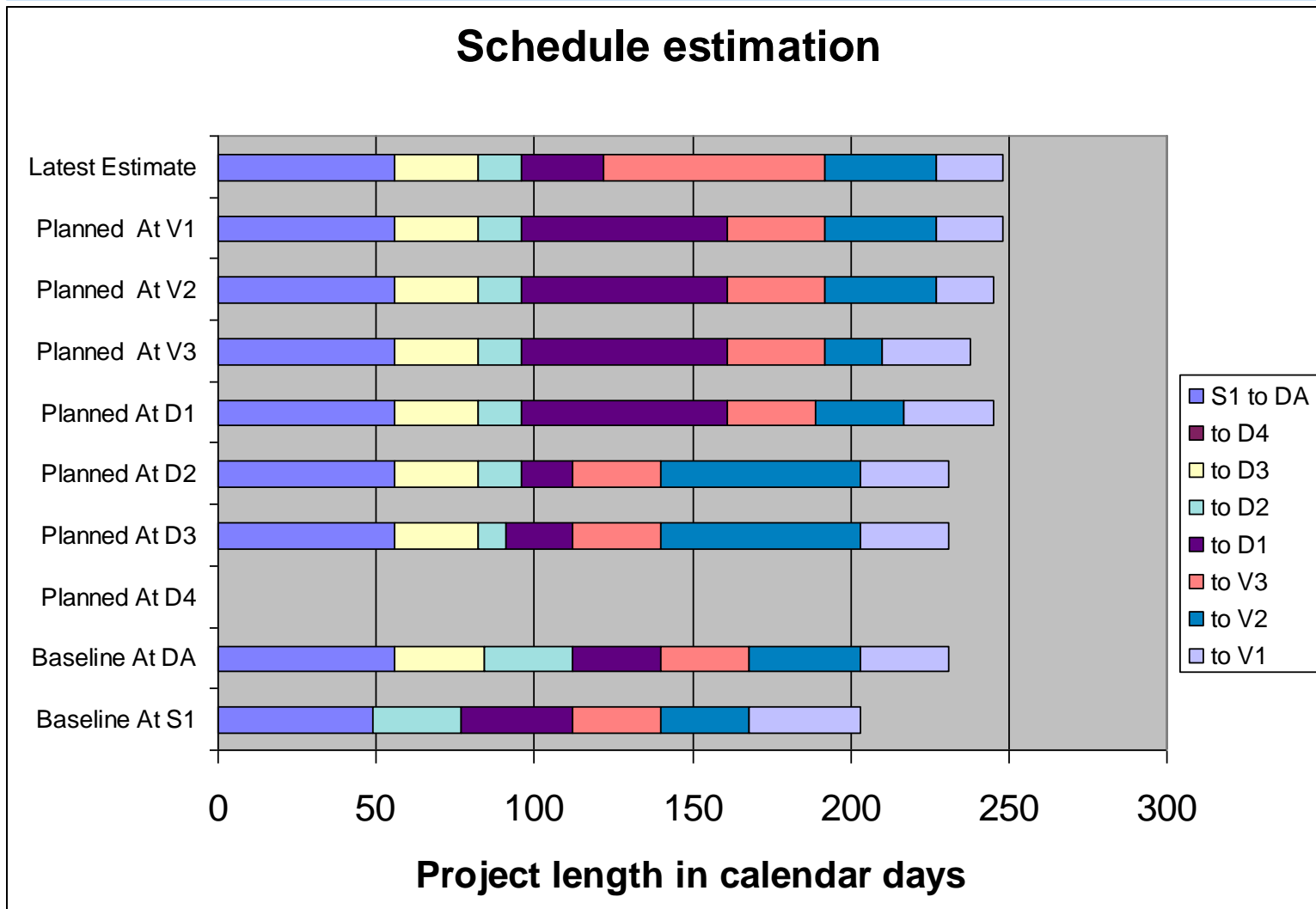
Agenda

1. Case Company Background
2. Initial attempts to change the game
3. Swinging the pendulum
4. Initial results
5. Recent development
6. Summary

Our Old Development Process from 1999

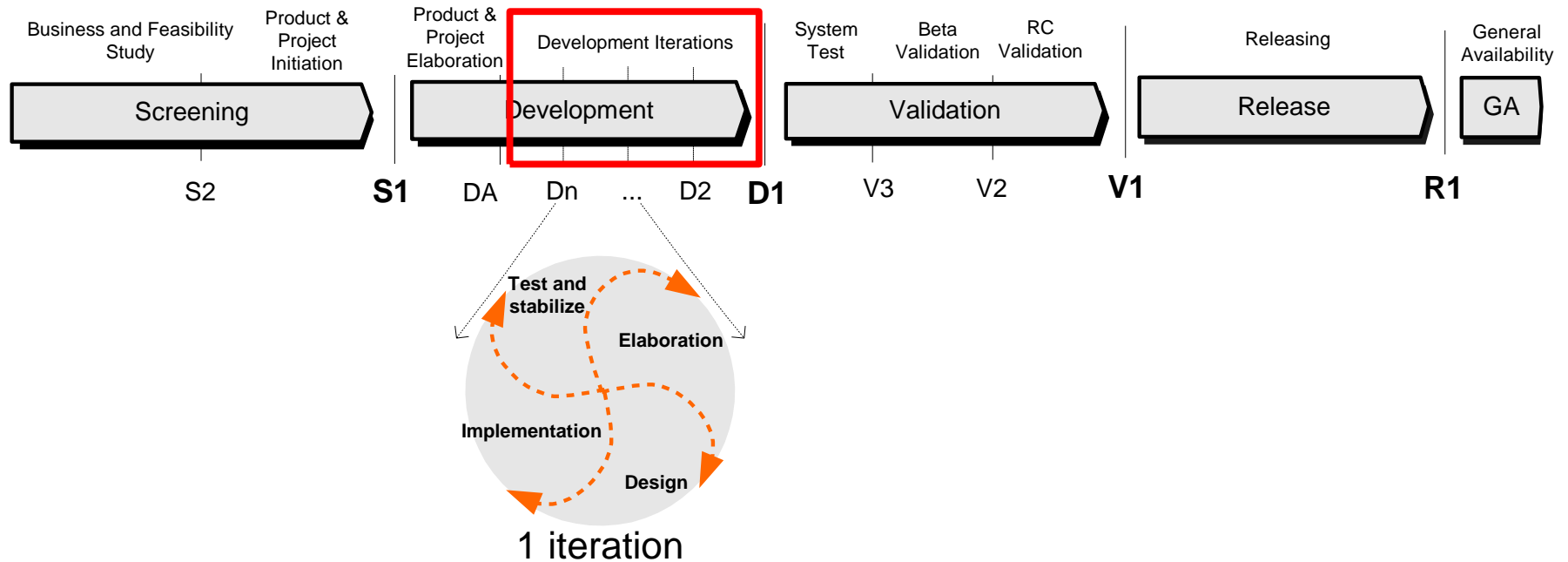


An Example of a Project During The “FPRP Era”



First Attempt to IID - FPRP 2.0

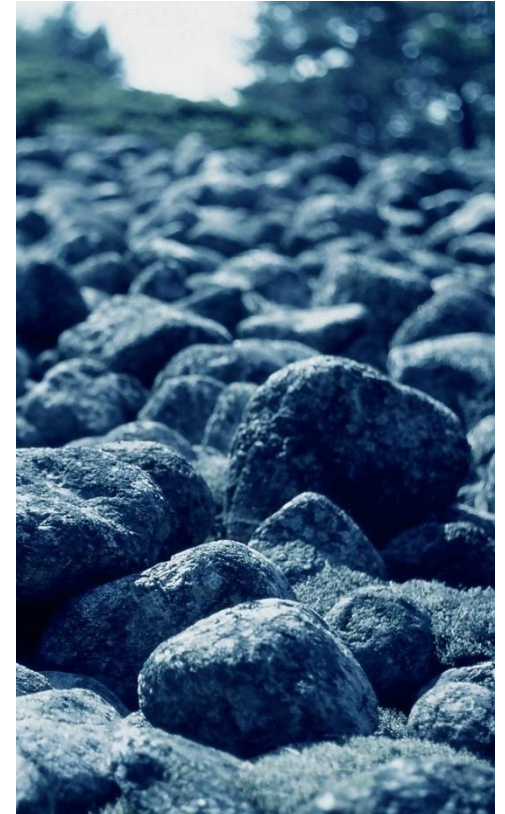
- more emphasis on iterations



- Incremental development of software and services
- PSG defines Dn gate criteria bases on checklists.
- The product under development should at all times be stable

The Burden of History Was Still Too Heavy

- No matter the changes, the **tendency** was still **towards waterfall**
- Customers may forget a missing feature quickly, but everyone will remember the **missed deadlines** for long
- **Not really knowing what customers really wanted.** We started to get more qualified feedback only when customers & partners had access to the software
- Delays in project schedules have **cascading effects** in larger organizations



Agenda

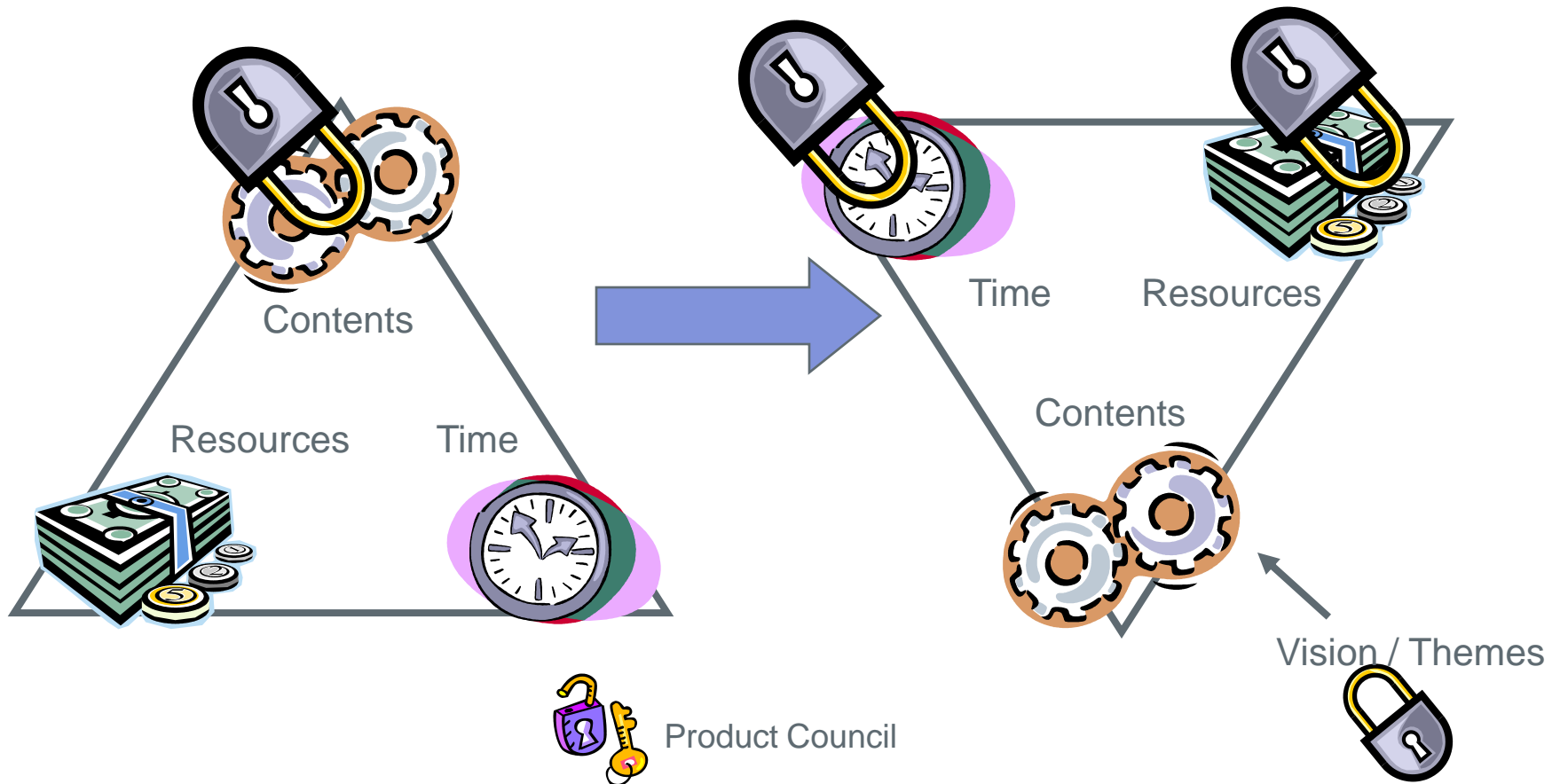
1. Case Company Background
2. Initial attempts to change the game
3. Swinging the pendulum
4. Initial results
5. Recent development
6. Summary

Autumn of 2003

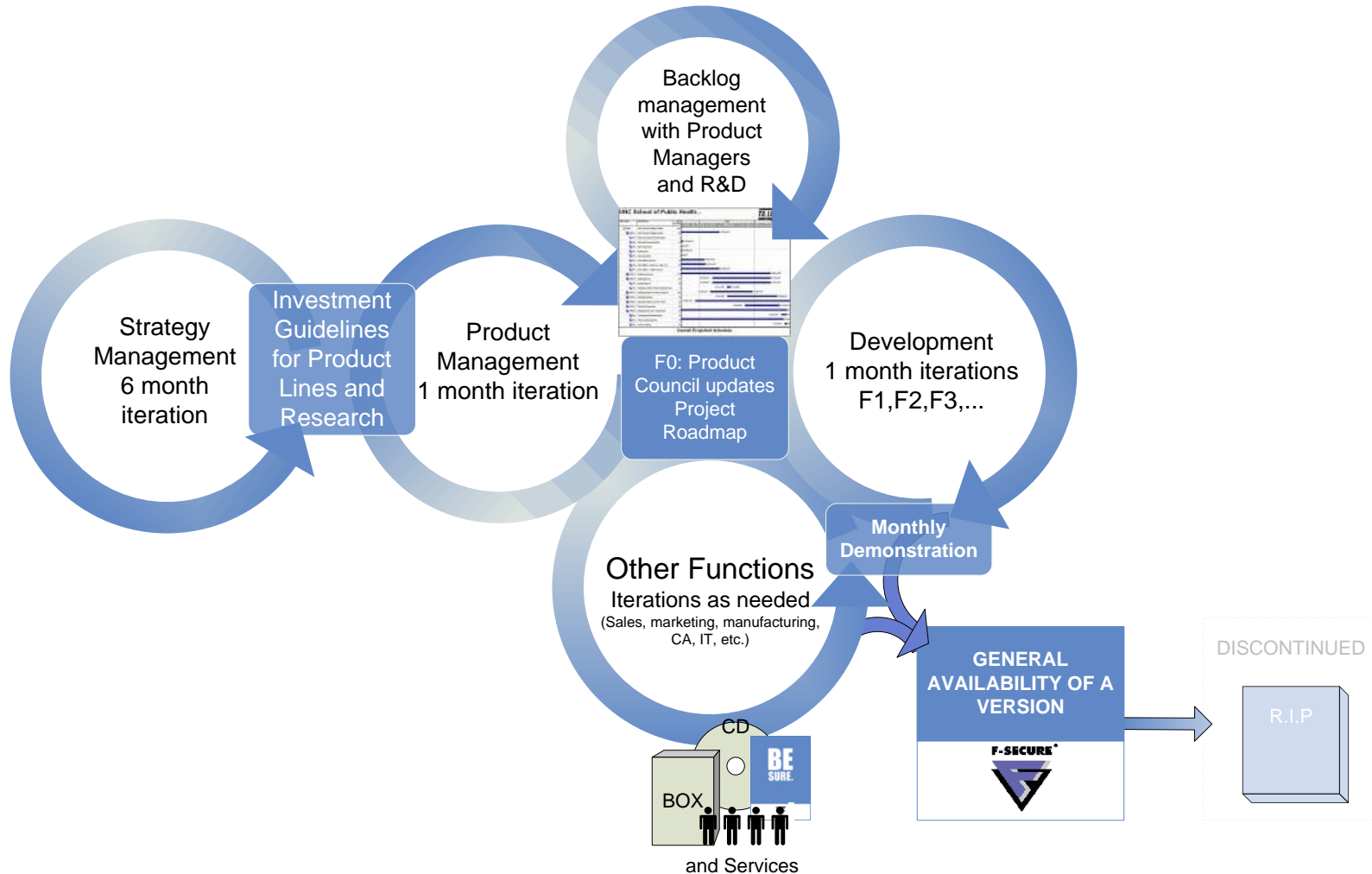
- Leap of faith to try something completely different

- Acknowledging the facts, that we had:
 - Two very different high level objectives:
 1. **Optimizing time (frequent deliveries) and fit to market (customer value) (Agile)**
 2. **Optimizing the cost to market across different customer segments and businesses (shared technology stack, platforms and product lines)**
 - Large and multiple teams involved in bigger projects and no answers from anyone how to manage agile in larger scale
 - Everyone not located in the same office, let alone the same country
 - No full picture how to scale agile into large scale, but just vision and faith in the direction

Change to Budget & Theme Driven Projects



Initial view on agile at the company level



Biggest differences in FPRP and FLEX

F-PRP (and RUP to an extent)

- Role & document focused
- Heavy emphasis on testing during validation
- Committing to the scope early in the process
- Fixed resources and scope translating flexible schedule (=missed deadlines)

FLEX (and Scrum to an extent)

- Team and customer value focused, deliverable-oriented
- End-to-end system testing during every iteration
- Committing initially to themes & minimal viable scope, final detailed scope through the iterations
- Fixed cadence & fixed resources; flexibility in scope (but also in time in rare cases, if deemed necessary)

Roads to Process Implementation



Deployment Strategy

SCOPE OF CHANGE

HIGH



LOW



LOW

HIGH

PACE OF CHANGE

Swinging the Pendulum And Jumping To Cold Water

- Starting with a few “pilot projects”
 - Catalyzing the change by completely removing the old process model and introducing the new one
- The big bang approach ("attack weeks")
 - Stopping the development for two weeks to “make room” for change
 - Training product managers, project manager, project teams to Scrum
 - Seen as necessity, as initially not much “pull” for change.

The Big Bang approach

1. Created needed instability to make a drastic change

- Not allowing to revert back to old way of doing or
- slowly trying to accommodate the change

2. Created confusion

- However, it forced us to prioritize & incrementally solve the problems that surfaced

=> A more gradual approach could have been easier, but wouldn't most likely produced the same results and could have been less productive



Key catalysts for change

1. The results and experiences of the pilot projects
 2. Teams that were already using similar methods within the old framework
 - Teams that had started the transition on their own before the change
 - Delivering "always running software" and "deliver early and often"
 - The scope management (although not in "backlog way") was already being done in the most successful projects
 - Customer involvement had already become a necessity for many of the projects
- => In retrospect agile was a natural change for some of our teams, which made the change easier

Cultural change brought by the Agile transition

- People started to be more accountable and taking more responsibility at every level, as they now feel a stronger ownership of their own work
 - More people focus on product design as opposed to executing single tasks distributed by the project manager
 - Many of the values in agile (customer collaboration, individuals and interaction) are enablers for innovation
- => Better commitment, communication and more people involved in innovation



Agenda

1. Case Company Background
2. Initial attempts to change the game
3. Swinging the pendulum
- 4. Initial results**
5. Recent development
6. Summary

Successes (1/2)

- **We were able to change to "truly iterative" process framework**
 - Short feedback cycle in all areas: product quality, process and procedures quality, scope
 - Quick decision making (wrong decision is, in many cases, better than no decision)
- **We have continuous integration and a stable build at all times in most of our teams**
- **We have the pressure to do the right things**
 - Product owners now have the pressure to choose what adds the most value to the customer since they have better visibility to what can and cannot be done in one sprint
 - After every sprint the teams are able to demo what is done and they have their retrospective in which they reflect how they can become even more efficient
- **We are focused on managing the scope**
 - Better focus on delivering the most valuable items/features first, so we are able to get the best ROI possible

Successes (2/2)

- **Risk management for projects is now part of the daily work**
 - we have daily meetings (however, not in all teams)
 - we have detailed planning events (sprint planning)
 - we react fast to risks that have materialized or have a high probability
- **Emphasizes the wellbeing and motivation of the team members to create better result**
 - Sustainable phase, helps people create value every day in the work
 - Replacing the intensity peaks at the end by a healthy doses of daily pressure
 - Pre defined rhythm encourages the team members to have fun and build a good team spirit
- **Teams can identify unnecessary tasks that do not bring any value and drop all such tasks**
 - Focusing on the value adding tasks people can also get a lot of achievements every day that they can feel good about.
- **We have been able to include localization and documentation into incremental and iterative**

Not all good, without challenges

- initial problems that surfaced

- Typical change resistance
- The entire company didn't change and adapt to agile way of working
- No good agreement on how roadmaps and related communication would be addressed
- Good culture of consistent documentation as the word comprehensive was translated to no documentation in many cases
- At the beginning week-end "sprints" as an "extension" to resourcing



Further Challenges in Latter Phases

- After we started to master the sprint and team level

- Frustration from other parts of the organization that R&D would not “commit to anything”; continuous reduction of scope
- Understanding of release goals and objectives not on sufficient level at project start and hard to get any commitments beyond the next iteration
- Too much initially focus on Scrum and not paying enough attention (and assigning business value) to engineering disciplines => building quality debt
- Longer term architecture just “emerged” – wasn’t coordinated and created architectural “silos” => building architectural debt
- Pain in creating completely new things, especially large systems as front end planning was seen bad by “religious agilists”
- Synchronizing & coordinating larger multisite efforts not effective

Agenda

1. Case Company Background
2. Initial attempts to change the game
3. Swinging the pendulum
4. Initial results
5. Recent development
6. Summary

Areas we have worked during the last year (1/2)

- Extending the process frame from point product release “projects” to solution releases (F-LEX 2)
 - Delivery of the full solution (including software, services, training, support, etc.) through one coordinated efforts
- Applying lean
 - Restructuring some team compositions to feature teams
 - Reducing handover waste between teams
- Reinforcing engineering disciplines through out teams
 - Continuous builds
 - Unit/module level testing
 - Code reviews
 - Etc...

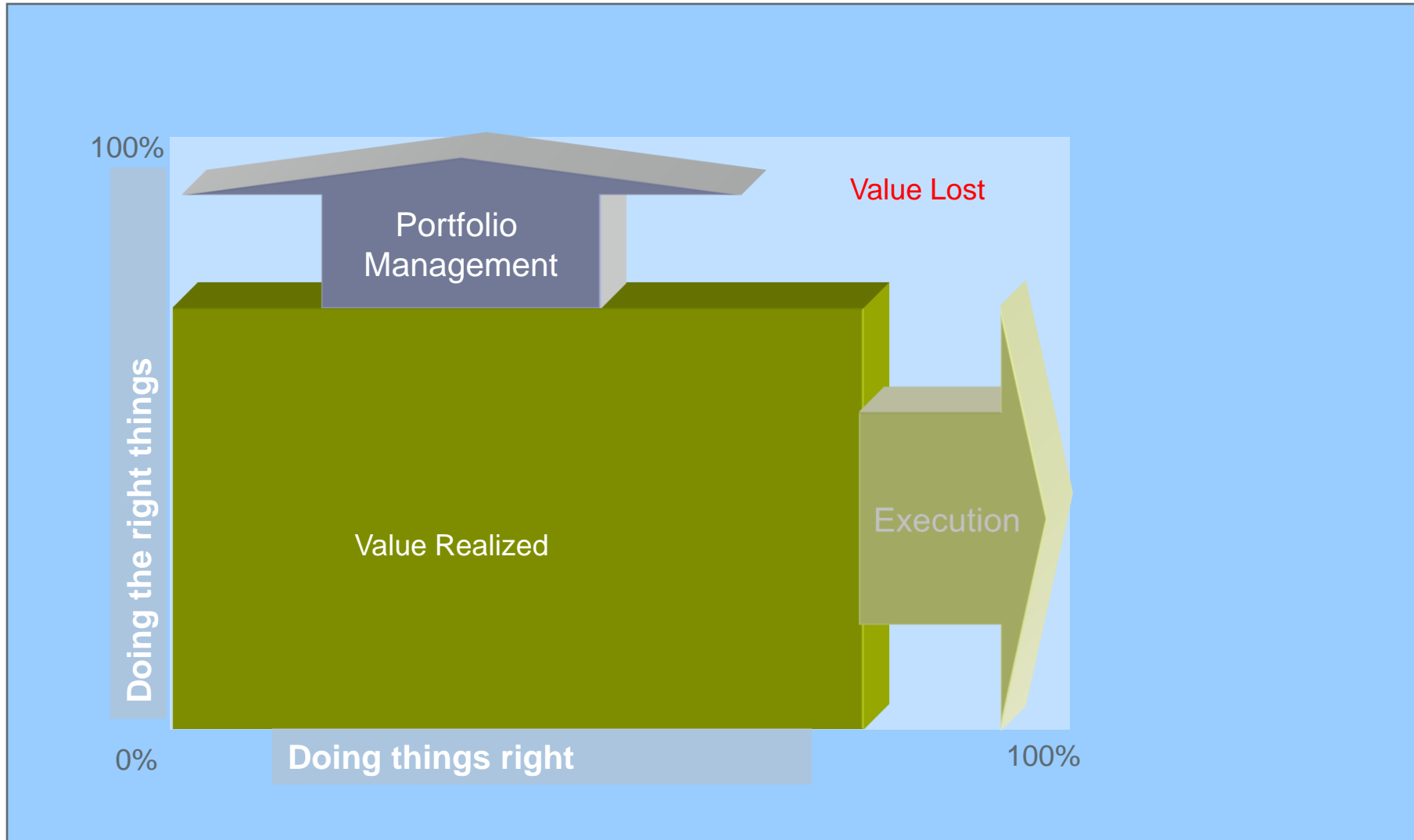


Areas we have worked during the last year (2/2)

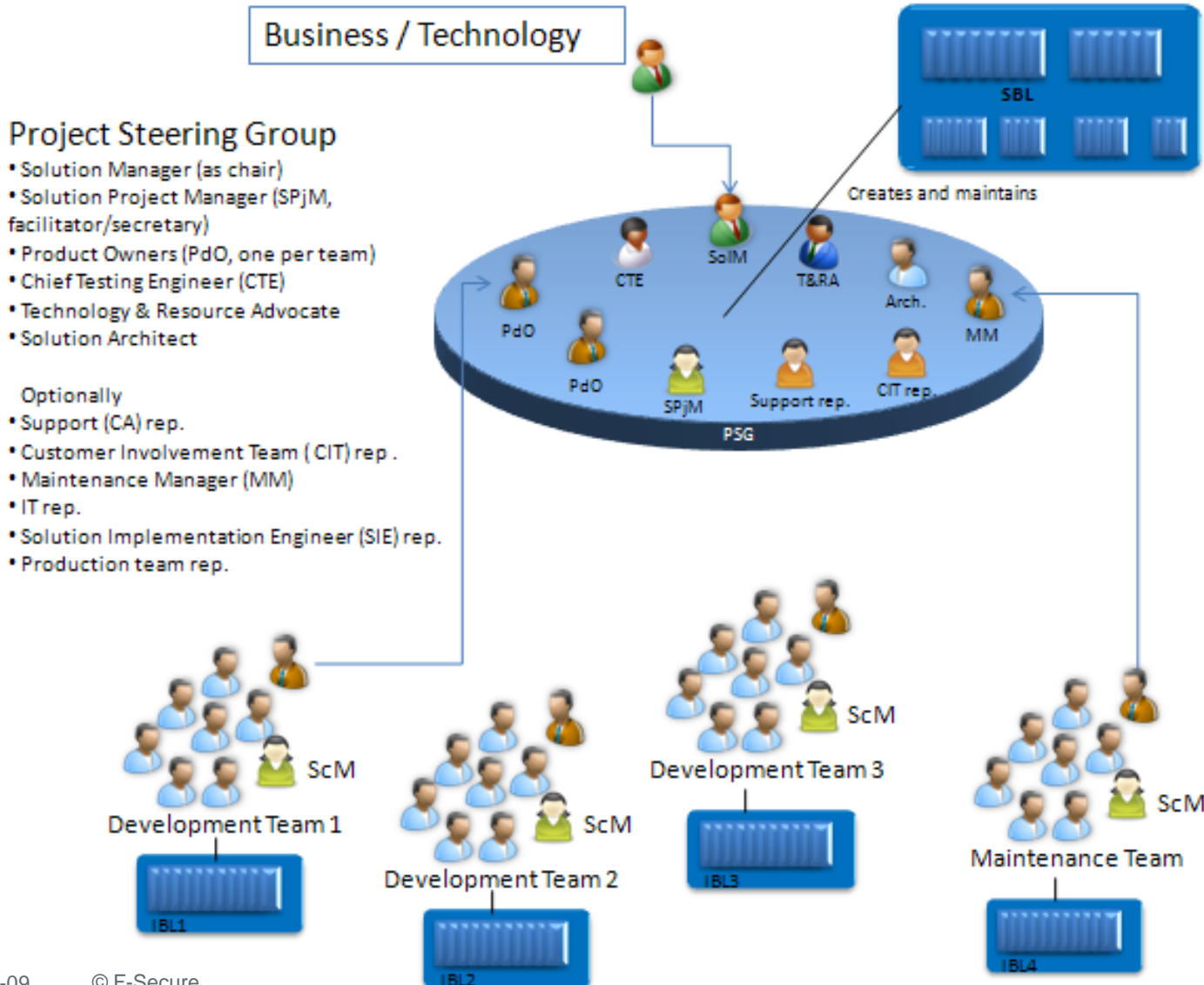
- “Internal open source” model
 - Everyone can contribute to common technology stack to remove bottle necks easily created with shared libraries and technologies
 - “Code guardians” that ensure a solid official trunk
- Corporate wide longer term architecture planning and implementation
 - Longer term architecture planning (architecture runways) and way to take the architecture research spikes and implementation to the iterations
- Scaling agile, lean and requirements management at the corporate level
 - Strategy, business planning and agile
 - Longer term portfolio level, capacity & investment planning
 - Solution vision, concept creation & release planning
 - Further clarifying “enterprise” & agile roles

Optimizing the efficient value delivery

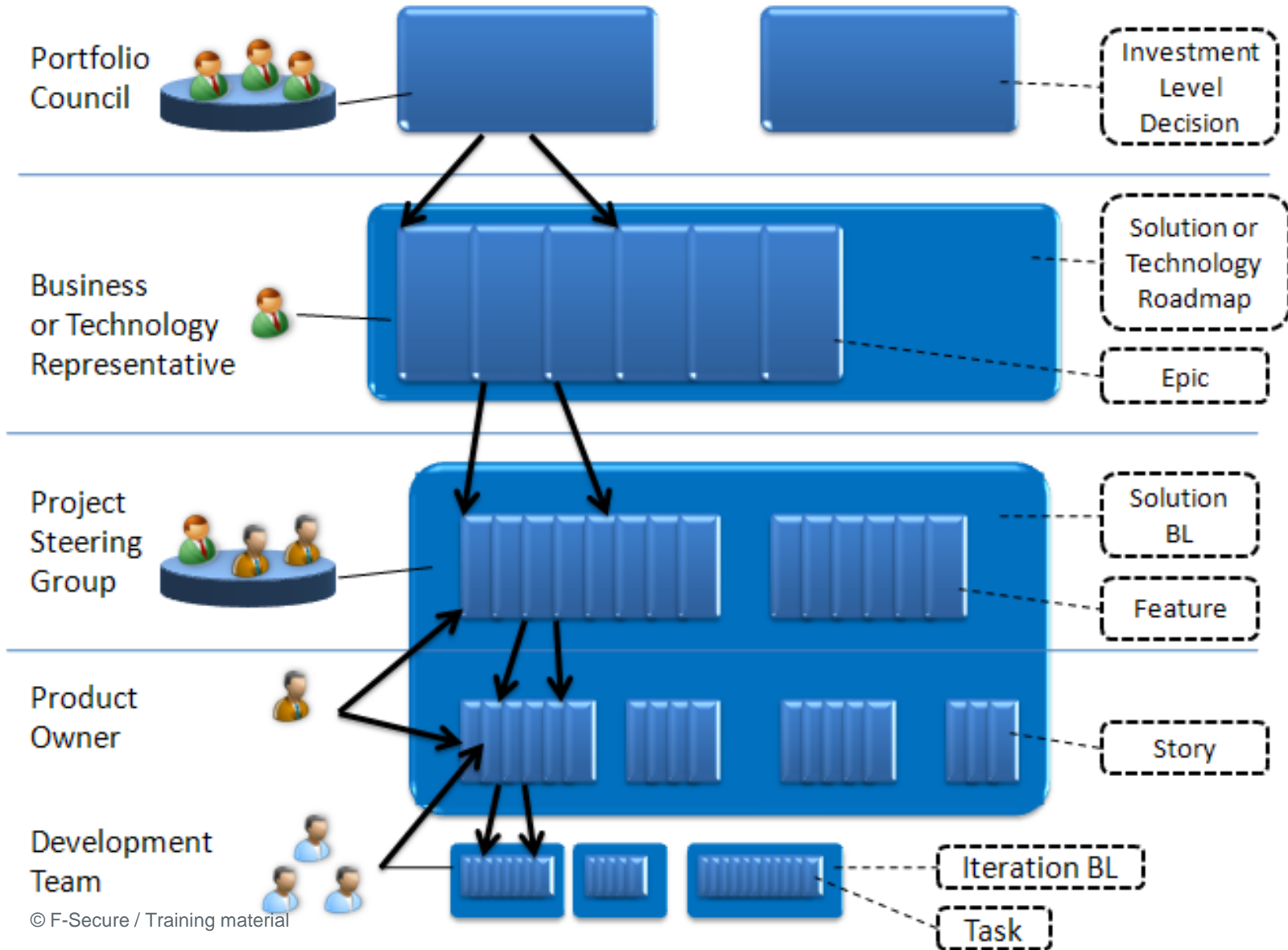
- Improving portfolio management besides execution



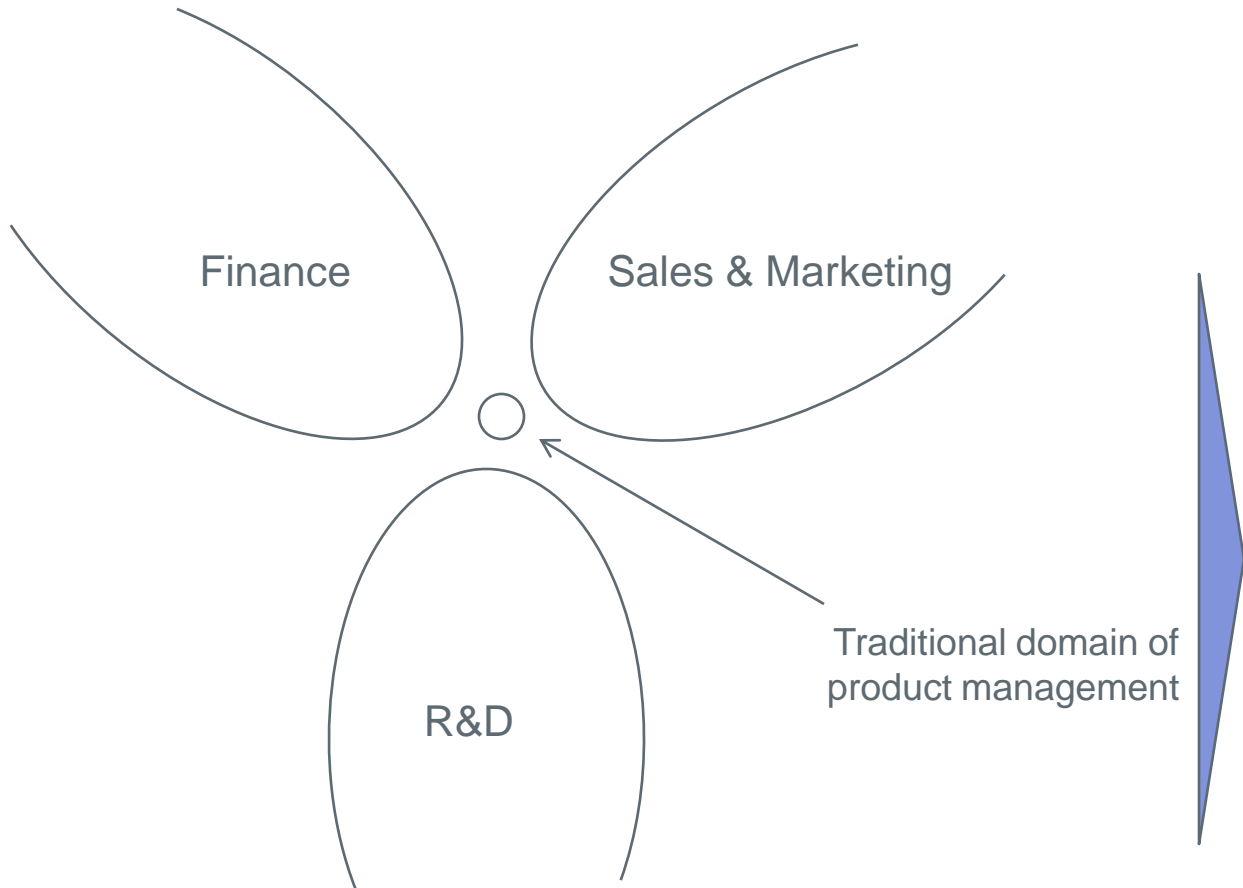
Project Organization in a Multi-Team Environment



Solution BackLog Work Levels

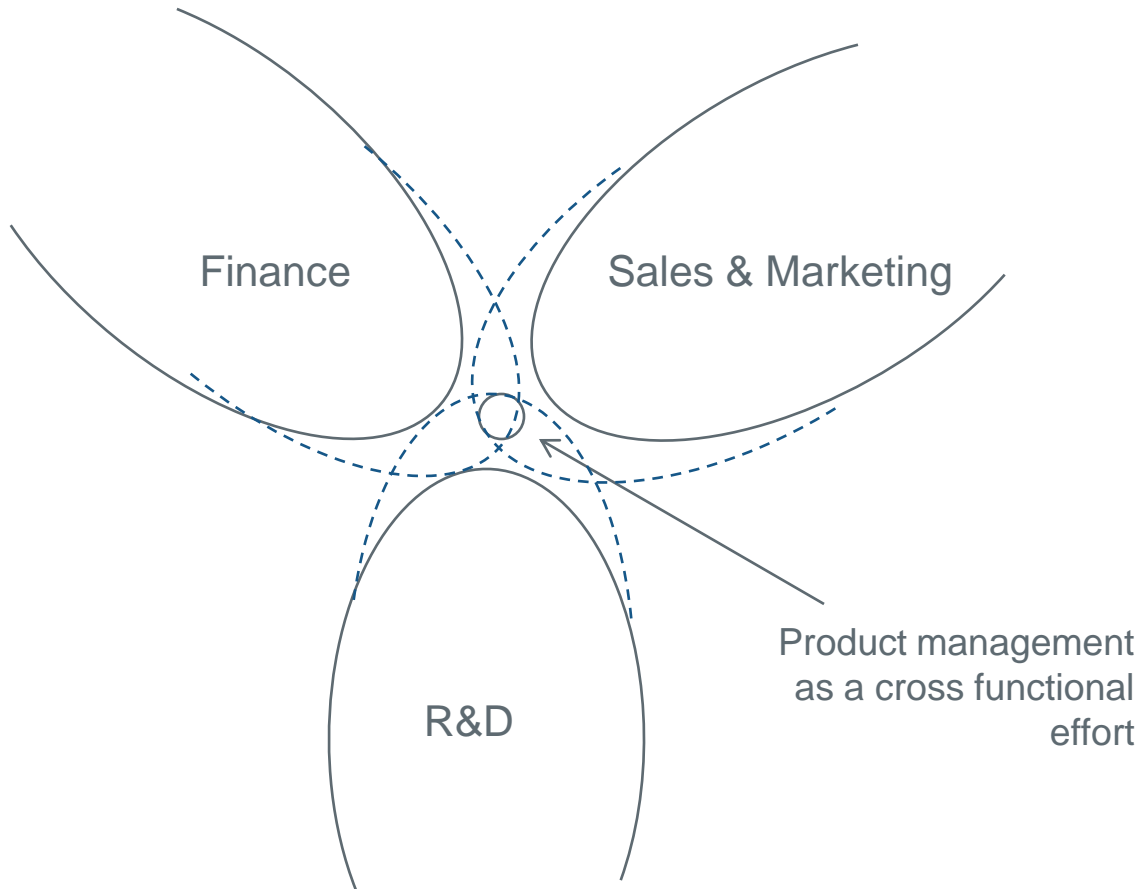


Traditional product management pulled to many directions



- Bridges the different “sub-cultures”
- Need to have capability to work together with different “cultures”
- Being “pulled” to many directions
- Hard to make best use of time

Bridging different “sub-cultures”



- Involving more people in understanding the different “sub-cultures”
- Product management consists of multiple people in different roles
- Defining product management discipline in more granular roles

Different roles in product management and linking them to positions and roles

<p style="text-align: center;">Market Expert</p> <ul style="list-style-type: none">• strategic and outbound• owner of price, place, promotion and longer term vision for product and release themes• understand market landscape, competitors and dynamics• product marketing and positioning• 80% external / 20% internal	<p style="text-align: center;">Product Expert</p> <ul style="list-style-type: none">• technical and tactical• owner of details in release implementation• understand the details of products (own and competitors)• drives the competitiveness of releases• 20% external / 80% internal
<p style="text-align: center;">Advocacy Expert</p> <ul style="list-style-type: none">• Advocates the existing solutions to the field• Helps sales and account management• Funnels back future requirements• 90% external / 10% internal	<p style="text-align: center;">Technology / Architecture Expert</p> <ul style="list-style-type: none">• Develops the vision for longer term technology competitiveness and operational efficiency• Drives daily implementation of sound architecture and technology choices• 10% external / 90% internal

Agenda

1. Case Company Background
2. Initial attempts to change the game
3. Swinging the pendulum
4. Initial results
5. Recent development
6. Summary

In retrospect... our lessons learned

Scrum is not enough and won't solve the problems at the enterprise level alone

Ensure good engineering disciplines. We started with Scrum and XP as the baseline, but forgot to put emphasis on the latter in the beginning.

Don't become religious. Some front-end planning and long term planning is still good and even needed especially with multi teams.

... Lessons learned

You need to have good agile teams to continue to the enterprise level agility

There is certain Maslow's hierarchy in building an agile enterprise.

Build gradually and ensure to have the right things in place first.

... Lessons learned

Don't expect that change is fast

Agile methods are simple to explain and simple to take into use, but the cultural change and company wide adoption takes time.

Now there is better knowledge and support in the industry however!

... Lessons learned

Ensure top management support and buy-in

The change to agile requires changes in the entire organization thus requiring top management buy out from all functions in the organization



The saga continues...

Our change started six years ago and still goes on...