Software Product Line Engineering
Processes and SPL
Organizational Issues
SPI/SPA

Tony Gorschek - tony.gorschek@gmail.com
Tony Gorschek

- Engineer / Problem Solver / Researcher
- PhD (Tekn. Dr.) Software Engineering, B.Sc.
- Business
  - 10+ years in industry (5 start-ups, CTO, Consultant, Chief Architect)
  - 8 years as a researcher

Research Areas
- Technology Product Management, Strategic and Value-based Product Development, Requirements Engineering, Process Assessment/Improvement, Quality Assurance, Practical Innovation
- Domains/partners: ABB (CRC, Substation Automation, Robotics), Ericsson (Charging), IBM (Tools and Dev. Support, Daimler (R&D), Volvo PV, Sony Ericsson, Danaher (AGVs))
Processes and SPL

Business

Economics
Planning
Strategy

Architecture
Roles
Responsibilities

Relationships

Organisation
People
Structures

Techn.
Processes

- Software Engineering Process: the total set of software engineering activities needed to transform requirements into software
- Product Development Process: the total set of engineering activities needed to transform requirements into products
  - Software (product) engineering refers to the disciplined application of engineering, scientific, and mathematical principles and methods to the economical production of quality software (products).
Process examples

- Requirements Engineering (Main Process Area)
  - Elicitation (Sub-process Area)
    - Task observation (Activity/Action)
- Configuration Management
  - Configuration Item Identification
    - Risk analysis
    - Volatility (change Prone) analysis
Process examples

- Requirements Engineering (Main Process Area)
  - Elicitation (Sub-process Area)
    - Task observation (Activity/Action)
- Configuration Management (MPA)
  - Configuration Item Identification (SPA)
    - Risk analysis (Action), Change Prone analysis (Action)
Processes and SPL and Organizations

- Business
- Organisation
- Architecture
- Process

- Economics
- Planning
- Strategy
- Techn.: Roles
- Responsibilities
- Relationships
- People
- Structures
Organization, roles and responsibilities

- Mapping of activities (actions) and process and roles to organization is critical as it is central to the successful realization and use of a PL
  - Amount of people working together (coherence within unit vs. collaboration btw units)
  - Accountability and funding
  - Decision hierarchy

- Why should we bother with this...

- Will people be able to see the product line and have the product line mindset?
- Local profit optimizations (e.g. project over product)
  - Mean time to decision is long (too many people involved)
  - Same role distributed (same work done in several places)
Organization, roles and responsibilities

- Mapping of activities (actions) and process and roles to organization is critical as it is central to the successful realization and use of a PL

- Organizational SIZE is crucial as it speaks to the impact of the organizational structure and the role and responsibilities division on the product line...

  Small organization has “closeness” and familiarity that can compensate for inadequacies, LARGE organizations DO NOT

“not my job”

Personal mind-set, and motivational structure plays a crucial role if a PL succeeds or not, much more so than having a perfect architecture or variability analysis

Imbalance in the organization (e.g. domination of application engineering over domain engineering)

What are individual engineers good at (like to do), skill set! E.g. Domain Eng. (high quality components and maintenance) vs. App. Eng. (build apps fast w. given components)
Funding: budget pressure... application units tempted to choose other company to provide domain (base)...
(especially initially when forming the PL, then after the domain part is so adapted to the apps that the apps can't find a better match)

Interactions: communication btw units –> overhead, addition of additional structure – can be compensated by accepting some overhead + formation of functional units
Process-Oriented Organizations

- Functional hierarchy is prime!
- Functional interaction is facilitated
- Flexible allocation of resources depending on need (btw application but also btw domain and application)
- People develop similar functionality for different products:
  - Easier to ensure integrity of architecture
  - Focus on reusability as it benefits you...
- more common in smaller organizations where communication is less of a problem

Main challenges:
- Different phases of engineering are not close
- Domain engineering spread out
- Communication btw units and planning is necessary
- Accountability (especially for domain assets is not clear)
**Matrix Organizations**

<table>
<thead>
<tr>
<th>Requirement Engineering</th>
<th>Application -1 Engineering</th>
<th>Application -2 Engineering</th>
<th>Application -n Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Requirements Engineer</td>
<td>Application Requirements Engineer</td>
<td>Application Requirements Engineer</td>
<td>Application Requirements Engineer</td>
</tr>
<tr>
<td>Domain Architect</td>
<td>Application Architect</td>
<td>Application Architect</td>
<td>Application Architect</td>
</tr>
<tr>
<td>Domain Developer</td>
<td>Application Developer</td>
<td>Application Developer</td>
<td>Application Developer</td>
</tr>
</tbody>
</table>

**Compromise btw product and process focus**

**Main challenges:**
- Scattered focus
- Complex management
Process Evaluation and Improvement

A process
...a sequence of steps performed for a given purpose, e.g. the software development process (IEEE-STD-610)
...the set of activities, methods, and practices used in the production and evolution of software (SEI CMM)

SPI (CMM, ISO/IEC15504, ISO9000, MIL-STD-498, Trillium, the V-Model etc)

SPI(RE) (REAIMS, SPICE, CMMI, REPM, iFLAP)

Process improvement
Continuous, Small-steps
Evolutionary
Measurement points
Evaluation – choice – plan – execute - evaluate
Process Evaluation and Improvement

- Do more with less!
- Quality assurance
- Repeatability
- Measurability
- Performing better than your competitors
- Performing better than you did before...
- Customer satisfaction
- Going from individual HEROES to a CURAGEOUS ORGANIZATION (not afraid to better itself)
- Evolvement
- Effective use of the resources available

- Etc............Ok then, but how do we achieve this? ->
Process Evaluation and Improvement

Find out what the problems are!
- State-of-practice (official and unofficial)
  (if it a'int broke, don't fix it...)
- Use the knowledge and views of the organizations constituents to est. base-line and to identify improvement issues
Process Evaluation and Improvement

Model based
- Framework/Standard
  - according to model
  - Changes - follow model
  - What do we do vs Framework

Inductive
- Internal (extern) knowledge
  - open inductive improvement
  - Change - according to priority
  - What do we do vs What do we want to do
Process Evaluation and Improvement 2

**Model based**
- + external knowledge
- + pre-packaged
- + best practices
- - top down
- - fit (generic)
- - superfluous parts
- - priority set

**Inductive**
- + adapted to the organization
- + only what is needed
- + org. priority
- +/- learning process
- + up-down, down-up
- - internal knowledge
- - larger demands on internal commitment

CMM/CMMI  ISO  SPICE  QIP  PDCA  iFLAP
Process Evaluation and Improvement

People

Artifacts

A

Project

B

project artifacts

C

Line

D

system/tools

process documentation

manuals

e tc

“Triangulation of Results”
Process Evaluation and Improvement

**OBSERVE!**
- Techniques are the same as for RE (e.g. reading documentation, interviews, brainstorming etc)
- You can use basically the same method for ELICITATION (est. knowledge about the domain, processes etc that the a system being developed is to support)
Elicitation techniques

- Interviews
  - Getting to know the present (domain, problems) and ideas for future system
  - Hard to see the goals and critical issues, subjective

- Group interviews
  - Stimulate each other, complete each other
  - Censorship, domination (some people may not get attention)

- Observation (Look at how people actually perform a task (or a combination of tasks) – record and review…)
  - Map current work, practices, processes
  - Critical issues seldom captured (e.g. you have to be observing when something goes wrong), usability issues seldom captured, time consuming

- Task demonstrations (Ask a user to perform a task and observe and study what is done, ask questions during)
  - Clarify what is done and how, current work
  - Your presence and questions may influence the user, critical issues seldom captured, usability problems hard to capture
Elicitation techniques 2

- Questionnaires
  - Gather information from many users (statistical indications, views, opinions)
  - Difficult to construct good questionnaires, questions often interpreted differently, hard to classify answers in open questions and closed questions may be too narrow...

- Use cases and Scenarios (Description of a particular interaction between the (proposed) system and one or more users (or other terminators, e.g. another system). A user is walked through the selected operations and the way in which they would like to interact with the system is recorded)
  - Concentration on the specific (rather than the general) which can give greater accuracy
  - Solution oriented (rather than problem oriented), can result in a premature design of the interface between the problem domain and the solution

- Study present systems/processes
- Study tools/techniques
- Ask for complementary material during sessions... and follow up!
Process Evaluation and Improvement

**Stakeholders**
- Stakeholder identification
- Stakeholder selection (access, representative?)

**Artifacts**
- Identify artifacts (project, line)
- Select (access, representative?)

- project plan
- SRS
- Roadmap
- Design
- Mapping (traceability info)
- process desc
- review documents
Family Evaluation Framework (FEF)

- Focuses on the evaluation of product lines (focus on aspects relevant to PLs)

- FEF should be used to evaluate product line organizations (or product line “like” organizations...)

Family Evaluation Framework (FEF) 3

- **Business**: business involvement in the SPL engineering and variability management. Business relationships between domain and application engineering, and the cost, profits, market value, and planning of variability.

- **Architecture**: domain and application architecture relations and how they are related via variability.

- **Process**: process usage and process maturity (use e.g. CMMI)

- **Organization**: effectiveness and distribution of domain and application engineering over the organization. Coordination, communication, how well is the organization suited to PL engineering and to the company
Family Evaluation Framework (FEF) 4

Each dimension has aspects based on the maturity of the aspects... the dimension gets a rating...

- Level 1
- Level 2
- Level 3
- Level 4
- Level 5

Basic

Advanced
Family Evaluation Framework (FEF) 5

- For each level FEF gives a characterization of the maturity for each aspect.

**Level 1**
- Financial: there is no, or little, involvement by the business. Systems are planned, sold, marketed on a single system basis.

**Level 5**
- Commercial: marketing and sales know the cost, profits, and ROI of SPLE and use this knowledge to improve business strategy.
Family Evaluation Framework (FEF) 6

- **Architecture**
  - Reuse
  - Ref. architecture
  - Variability

**Level 1**
- there is no or unsystematic reuse (not planned or controlled and systematized)

**Level 5**
- there is a systematic reuse based on an asset repository (asset under CM that is used for reuse)
Family Evaluation Framework (FEF) is used to evaluate the processes used. FEF uses parts of CMMI (and Level 1 in FEF does not always correspond to CMMI Level 1!).

http://www.sei.cmu.edu/cmmi/
Family Evaluation Framework (FEF) 8

Balance, one dimension influences the other...

---

Diagram:

- Organisation
- Process
- Business
- Architecture

Dimensions:

1. weak connected
2. managed
3. managed
4. variant products
Case study

- Do the evaluation (or suitability analysis) according to relevant framework (see ass. desc.)
- The interview questions, design (e.g. selection of whom you talk to) and how these questions relate to the framework should be mapped.
- The subjects answers (raw data) should also be turned in (appendix).
- Your interpretations of the answers should be a part of the report, e.g. why you judge a certain level.
- Some aspects are more suited to other data sources than interviews, but you may use interviews. Bonus if you use triangulation (e.g. confirm in other sources, e.g. two interviews or one interview and documentation)
  - E.g. ask about reuse, get an answer that indicated Level 5, then you look at their asset management and control that the opinion of the interview subject corresponds to reality.
  - E.g. 2: ask two different developers (separate interviews) about reuse, compare answers.
- The interviews you design should be semi-structured to reflect FEF, but do not be leading. Ask follow-up questions to be sure you understand enough to make judgement.
Case study

- ... so what is your status?
- ... practical tips...
  - first contact...
  - getting resources...
  - doing the work...
  - reporting (not only for the course...)
Case study

- ... so you got a lot of information about how the company works...
- ... and then what?
  - mapping/evaluation (to FEF/BAPO)
  - prioritize and order (prio and dependency)
  - create an improvement plan (stepwise)
  - report it...