Pattern Language for Project Management in Global Software Development

Antti Välimäki
Metso Automation / R&D

23.8.2011
This is Automation
We serve you in a wide range of industries

- Power
- Pulp and paper
- Oil and gas
- Industrial gas
- Chemical and petrochemical
- Other customer industries
A wide scope of automation and information technology

Automation systems  Advanced process controls  Information management  Condition and runnability monitoring  Quality control systems

Automation solutions

Profilers  Automated on/off valves  Intelligent control valves  ESD valves  Analyzers, measurements and laboratory
Metso DNA Architecture

Operation, Maintenance, Reporting

Star or ring topology redundant Ethernet network

Remote connections

Reporting, Enterprise integration

Controls, Optimization, Connectivity

Centralized

Distributed

Safety instrumented system

XML

Engineering & Maintenance, Asset Management, Information services

Office

Office

Office

Field

Field

Field

© Metso
Metso Automation toimipisteitä lähellä asiakkaita
Globaali ohjelmistokehitys (GSD)

Globaali ohjelmistokehitys on malli, jossa eri osapuolet

- erilaisisista kansallisista ja organisaatiokulttuureista
- erilaisista maantieteellisistä sijainneista käsin ja
- usein eri aikavyöhykkeiltä käyttävät tiedonjako- ja kommunikointiteknologioita toteuttaessaan yhteistyössä haluttuja ohjelmistoja
Millaisia etuja globaaliin ohjelmistokehitys voisi tarjota?

- Parempi ymmärrys asiakastarpeista
  - Voidaan olla lähellä asiakasta
  - Voidaan kehittää ohjelmistoja paikallisten henkilöiden kanssa
- Parantaa kilpailukykyä paikallisilla markkinoilla
- Yritysostojen parempi hyödyntäminen
- Lahjakkuuksien hyödyntäminen maailmanlaajuisesti
- Kustannussäästöt
- Nopeuttaa kehitystä, kun töitä voidaan tehdä osin rinnakkain ja suuremmalla henkilömäärällä
- Hyödyntää muualta löytävää tietämystä
Millaisia haasteita globaali ohjelmistokehitys tuo mukanaan?

• Haasteita ovat esimerkiksi katkokset projekteihin liittyvissä
  - kommunikoinnissa
  - koordinoinnissa ja
  - kontrolloinnissa

• *Kommunikointi* on tehokkainta kasvokkain, kun kuullaan äänen sävyt ja sanojen painotukset, ja nähdään myös kasvojen, käsien ja vartalon liikkeet
  - Hajautetussa ympäristössä luetaan usein vain sähköposteja tai kuullaan yleensä vain ääni, jolloin suuri määrä informaatiosta jää välittymättä
Miten paljon tietoa välittyy eri kommunikointimenetelmillä?
Millaisia haasteita globaali ohjelmistokehitys tuo mukanaan?

- **Koordinointi** on toimintaa, jolla yhdistetään eri tehtävien tekeminen kohti yhteistä tavoitetta
  - Kun toimitaan hajautetusti, on vaikeaa tehdä nopeita muutoksia projektin jäsenten toimintaan, koska etäisyys ja mahdolliset aika- ja kulttuurierot hidastavat vaikutusmahdollisuksia
  - Ongelmien selvittäminen on hitaampaa, koska tarvittavia henkilöitä ei saada keskustelemaan kasvokkain niin nopeasti kuin keskitetyssä kehityksessä

- **Kontrollointi** on valvontaa, jolla varmistetaan, että projektissa noudatetaan sovittuja menetelmiä ja tavoitellaan sovittua laatutasoa
  - Hajautetussa toiminnassa projektipäällikön on vaikea valvoa toisella paikkakunnalla olevan ryhmän toimintaa
  - Koska valvonta on vaikeampaa, ongelmat huomataan vasta myöhemmin, jolloin niiden korjaaminen on usein työläämpää ja projekti hidastuu entisestään
Millaisia haasteita globaali ohjelmistokehitys tuo mukanaan?

<table>
<thead>
<tr>
<th>CCC/Distances</th>
<th>Temporal Distance</th>
<th>Geographical Distance</th>
<th>Socio-cultural Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Reduced synchronous communication (1)</td>
<td>Face-to-face meetings difficult (4)</td>
<td>Cultural misunderstandings (9)</td>
</tr>
<tr>
<td>Coordination</td>
<td>Typically increased coordination costs (2)</td>
<td>Reduced trust (5) and A lack of critical task awareness (6)</td>
<td>Inconsistent work practices (10) and Reduced cooperation (11)</td>
</tr>
<tr>
<td>Control</td>
<td>Management of project artifacts (3)</td>
<td>Difficult to convey strategy (7) and Low-cost “rivals” (8)</td>
<td>Different perceptions of authority (12) and Managers must adapt to local regulations (13)</td>
</tr>
</tbody>
</table>
Miten haasteita lähdetty ratkaisemaan?

Esitelmään liittyvässä väitöstutkimuksessa on haettu ratkaisuja kysymyksiin:

- Kuinka hyviä projektinhallintakäytäntöjä pitäisi kuvata?
- Kuinka löydetään hyviä projektinhallintakäytäntöjä globaaleista ohjelmistokehitysprojekteista?
- Mitkä ovat hyviä projektinhallintakäytäntöjä globaaleissa ohjelmistokehitysprojekteissa?
- Kuinka tällaisia käytäntöjä voisi evaluoida?
A Template of a Process Pattern

<table>
<thead>
<tr>
<th>Name:</th>
<th>The name of the pattern.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem:</td>
<td>A brief description of the problem.</td>
</tr>
<tr>
<td>Initial context:</td>
<td>The situation to which the pattern solution applies.</td>
</tr>
<tr>
<td>Roles:</td>
<td>The roles implementing the pattern.</td>
</tr>
<tr>
<td>Forces:</td>
<td>Forces that affect the situation.</td>
</tr>
<tr>
<td>Solution:</td>
<td>The required instruction to solve the problem in the context.</td>
</tr>
<tr>
<td>Resulting Context:</td>
<td>The situation/context which will result from performing the pattern solution.</td>
</tr>
</tbody>
</table>
Case: GSD project management in Agile projects

- Case description
  - This case focused on two product lines consisting of several products that were partly developed in different sites.
  - The case focused on two teams each having 3 to 6 projects running in parallel.
  - The projects were geographically distributed over 2 or 3 sites in two countries.
  - The cooperation model was one virtual team which consist of employees of two to three different companies
  - Each project had typically less than 10 team members
  - The main results were
    - the process of mining agile project management process patterns and
    - a group of agile project management patterns in a GSD environment.
An example of a GSD pattern

<table>
<thead>
<tr>
<th>Name:</th>
<th><strong>GSD4- Divide and Conquer with Iterations</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem:</td>
<td>How to make a project plan which is manageable in a GSD project?</td>
</tr>
<tr>
<td>Initial context:</td>
<td>In the beginning of a project only the main features are known.</td>
</tr>
<tr>
<td>Roles:</td>
<td>Project manager.</td>
</tr>
</tbody>
</table>
| Forces: | 1. A big project plan is difficult to manage in a GSD project.  
    2. Difficult to know the whole contents and the work estimations of a project in the start of a project.  
    3. Visibility of project status is poor in GSD.  
    4. Possible new application architecture, technologies etc. are unknown. |
## An example of a GSD pattern

| **Solution:** | Project manager will split a project plan into several iterations.  
Implement the following actions:  
- Plan several iterations to describe the project plan because iterations are easier to control and it is easier to make changes to a plan.  
- Develop new application architecture and module structure in the main site during first iterations, if needed.  
- Explore the biggest risks (e.g. new technologies) in the beginning of a project.  
- The length of iteration can be e.g. 2-4 weeks to improve control and visibility. Main site can have 4 weeks iteration and other sites 2 weeks to improve visibility. |
| **Resulting Context:** | Iterations improve the visibility of a project and motivation of team members in a GSD project. Iterations also make it easier to control a project when you split the whole project into many manageable parts. Iterations also provide feedback and the possibility to learn from earlier iterations. However, administration work is increased with several iterations. |
Pattern Language for Project Management in GSD and PRINCE2

1. GSD Strategy
2. Fuzzy Front End
3. Collocated Kick-Off
4. Divide and Conquer with Iterations
5. Choose Roles in Sites
6. Communication Tools
7. Common Repositories and Tools
8. Work Allocation
9. Architectural Work Allocation
10. Phase-Based Work Allocation
11. Feature-Based Work Allocation
12. Common Processes
13. Iteration Planning
14. Multi-Level Daily Meetings
15. Iteration Review
16. Organize Knowledge Transfer
17. Manage Competence
18. Notice Cultural Differences
GSD Strategy

• **Problems:**
  - How to organize GSD in a company?

• **Solution Outline:**
  - List the reasons and motivation to start GSD-based development in a company.
  - Find out the competence and cost level of your own company, your partners and other related partners. *Manage Competence* pattern can be used with this action.
  - Make a SWOT (Strengths, Weaknesses, Opportunities and Threats) for a preliminary GSD strategy.
  - Make a short and long term plan to implement GSD strategy in your R&D, e.g. for make work allocation by *Work Allocation* pattern.
  - Make also a separate risk plan to manage risk with GSD. This is very important because GSD brings a lot of new risks to manage, e.g. intellectual property risk, loss of proprietary knowledge risk, product security risk, and infrastructure risk.
  - Plan measurements for the real costs of GSD.
Collocated Kick-Off

• Problems:
  - What is the goal of a GSD project and who are the members of a project? How to build trust between team members?

• Solution Outline:
  - Arrange a kick-off meeting for all relevant team members to meet face to face.
  - Everybody presents himself / herself.
  - Present common goal and motivation of this project.
  - Present release plan which is made by Divide and Conquer with Iterations pattern.
  - Present responsibilities of each site and team members, if possible. The result of Choose Roles in Sites pattern can be used with this action.
  - Briefly introduce tools and repositories which are chosen for a project by Communication Tools and Common Repositories and Tools patterns.
  - Briefly present common processes in a project which are specified by Common Processes pattern.
  - Also train cultural issues for team members according Notice Cultural Differences pattern.
  - Organize leisure activities to increase trust between team members.
Communication Tools

• **Problems:**
  - How to choose communication methods and tools in GSD?

• **Solution Outline:**
  - Have reliable and common communication methods and tools in every site.
  - Use of video conferences, web cameras, net meeting applications, chat, conference phones, Skype, mobile phones, electronic calendars, discussion tools, wiki tools etc. improves the efficiency of communication.
  - Use different tools at the same time as a net meeting application to show information and project data, conference phones to have good sound and chat tool to discuss in written form if there are problems to understand e.g. English language used in other sites.
  - Record e.g. meetings and key presentations to improve knowledge sharing. Records of meetings or presentations make it possible to listen to them once again, if needed.
  - Use common repositories to share artifacts, which can be implemented by Common Repositories and Tools pattern.
  - Publish availability of team members, common working times, holidays etc. to help communication and coordination of different meetings.
  - Also notice cultural differences which also affect communication methods and use of tools.
Common Repositories and Tools

• **Problems:**
  - How to choose communication methods and tools in GSD?

• **Solution Outline:**
  - Provide a common Application Lifecycle (ALM) Management tool for
    • all project artifacts (e.g. Product and Sprint backlog items, source code, development documents, faults descriptions etc.),
    • reports,
    • process guidelines (workflow, guidelines),
    • process information (e.g. who does what and when),
    • traceability (e.g. which information is related with each other),
    • communication tools (discussion forum, chat, visualize information, notifications to users etc.).
  - ALM tool can be implemented as a single tool set or it can be a group of different tools which has been integrated with each other.
  - Use different levels (team, project, and program) reports to improve visibility of status of projects.
  - Effective access right control i.e. user rights methods and role-based views in use to see certain data.
Analysis between GSD Pattern Language and Related Research

- GSD has brought many benefits but also challenges. Solutions to these challenges have been investigated by several authors like [Coplien and Harrison 2005], [Sangwan et al. 2006], [Paasivaara et al. 2010], [Woodward et al. 2010], [Šmite et al. 2010], and [Richardson et al. 2010].
- Subsection 6.1 discusses the similarities and differences between above studies and GSD Pattern Language.
- More information in doctoral dissertation: Pattern Language for Project Management in Global Software Development
Analysis between GSD Pattern Language and Related Research


Principal elements of Application Lifecycle Management [Kääriäinen 2011]

- Tool integration
- Communication
- Reporting of lifecycle artefacts
- Traceability of lifecycle artefacts
- Creation and management of lifecycle artefacts
- Process support

Requirements definition
Design
Coding and unit testing
Integration & verification
Release
Maintenance
Project management
Requirements management
Configuration management
Product manager
Project manager
Developer
Tester
Architect
Lessons learns

• Is agile project management (Scrum) suitable for us?
  - We are developing some of our products in a distributed way
  - we are going to continue to use more Scrum based practices, artifacts and roles, especially in GSD (Global Software Development) to improve visibility, feedback, communication, business value etc. although to implement Scrum requires a lot of work.
  - But when we are developing our system product (25 years old) also with collocated groups and we have adopted some practices from Scrum such as Product backlog method, but we have still Product and Project managers, Developers and System testers.

• Are our ALM tools suitable for us?
  - We have developed our ALM (Application Lifecycle Management) solution with Lotus Notes applications to support our system development and we are going to use and further develop these applications.
  - We are also using commercial ALM tools in special cases, but challenges with these tools are e.g. high costs of licenses, support etc. and integration challenges with our Notes based tools.
  - Use of ALM tools is very important both in collocated and in GSD.
References for related doctoral dissertations

• Pattern Language for Project Management in Global Software Development / Antti Välimäki

• Towards an Application Lifecycle Management Framework / Jukka Kääriäinen
Kiitos!