An SME Transition from Plan-Driven to Hybrid Project Management with Agile Software Development Methods

Stefan Biffl\textsuperscript{1} Richard Mordinyi\textsuperscript{1} Günter Raidl\textsuperscript{2} Heinrich Steininger\textsuperscript{2} Dietmar Winkler\textsuperscript{1}

\textsuperscript{1}TU Vienna, Institute of Software Technology, CDL-Flex, Austria
\textsuperscript{2}logi.cals GmbH, St. Pölten, Austria

http://cdl_ifs.tuwien.ac.at
Motivation & Goals

Motivation:
- Small and medium enterprises typically need to align plan-driven (heavy-weight) and agile (light-weight) software development processes.
- Main goal is to enable high flexibility (e.g., considering frequent changing customer requirements) aligned with a plan-driven approach (e.g., defined by contracts), i.e., some hybrid approach to benefit from both engineering processes.
- Process Improvement Initiative

Key research questions focus on:
- How to enable the alignment of plan-driven and agile engineering processes?
- What are the benefits of such a hybrid approach?

Goals of the paper:
- Concept of a hybrid project management approach.
- Lessens learned and success-criteria of a successful transition and application of the modified approach based on two case studies.
Related Work

- **Plan Driven Project Management**
  - Widely spread in industry because of defined plans.
  - Separation of individual phases (including quality assurance steps).
  - Require stable requirements with limited capability of changes.

- **Agile Project Management**
  - Growing importance in the last decade of software development.
  - High level of customer interaction and collaboration.
  - Flexibility regarding requirements changes.

- However, new product development projects typically require both, from research prototypes to industry products.*

---

Crisis with a Plan-Driven Approach (Project A)

- **Project Goal:** New product development, designed for safety-critical certification.

- **Challenges & Risks:**
  - New / unknown application domain and technologies.
  - Limited resources: 9 engineers
  - Estimated effort: 30 person years.
  - PM-Approach: Strictly plan-driven

- **Intermediate Management Review after 60% → project crisis:**
  - Project targets are moving away from completion.
  - Unexpected and frequent feature requests.
  - Inefficient project and process tracking and tracing.

- **Counter measures → make the best out of the situation:**
  - Changing the PM-approach towards agile.
  - Supported by external consultants (e.g., by using Kanban)

- **Results:**
  - The project could be finished successfully but with reduced functionality.
Changing the PM Strategy

- Focus on Best-Practices out of two worlds:
  - Plan-Driven Approaches: structure and basic roadmap
  - Agile and flexible Approaches: development sprints based on agile principles.

- Research Questions
  - How to combine best practices from plan-driven and agile PM approaches towards a hybrid approach?
  - How to demonstrate the benefits of the hybrid approach?
1. In the **plan-driven project structure plan** (PSP) the agile sprints have to be represented for planning, coordination, controlling, and measurement of progress;

2. The **process interface** between PSP and sprints has to be defined; and

3. In the **sprint backlog** the needs coming from other work packages in the PSP have to be represented for effective coordination.
Case Study with the Hybrid PM-Approach (Project B)

- **Project Goal**: Software Development Environment for Automation Systems Software Design & Development (Logi.cals Open 3).
  - Software Research and Development Project
  - Engineering system in a systems-of-systems multi-disciplinary engineering environment to develop industrial production plants.

- **Challenges & Risks**:
  - New / unknown application domain and technologies (again a challenge)
  - More but still limited resources: 20 engineers from different organizations
  - Estimated effort: 3 years with yearly major deliverables to key customers.
  - PM approach: hybrid project management
    - Plan-Driven top-level framework
    - Agile Sprints
1. **Plan-Driven PM.** Basic project management framework, e.g., technology exploration, training, concept development → stories/sprints.

2. **Parallel Sprints.** Individual sprints aligned with plan-driven work packages. Parallel sprints for software development, research prototypes, marketing → simplification of communication.

3. **Synchronization.** Needs coming up from sprint tasks get communicated to the PM and get planned in plan-driven WPs.

---

**Legend:**
- AP: Work Packages
- ES: Engineering / Development Sprints
- FS: Research Sprints
- VS: Marketing sprints

Institute of Software Technology and Interactive Systems
Balancing the Software Development Process

1. **Feature Map.** Epics and stories driven by marketing and research; concrete requirements or innovative ideas → dependencies become visible.

2. **Basic Features** are planned for shipment to the key customers → (Research) Prototypes.

3. **Dependencies.** Selected features sets for different versions of the product (different colors)

4. **Backlog** holding ideas as candidate for future development (not planned yet) → Foundation for Sprint planning.
1. **Sprint Planning.** Kanban boards, used by the development team, to organize the work tasks in sprints, showing the work load of resources and progress control.

2. **Plan-Driven progress control.** Kanban boards also provide for the project management progress control on task level from sprints.

3. **Management dashboard.** The data from the Kanban boards is aggregated in the bi-weekly project team meetings for controlling to allow the effective and efficient update of the management dash-board for reporting.

Applied Tool (Selection): Jira, Confluence, Continuous Integration & Test (Jenkins), Reviews (Gerrit)
# Project Risk Assessment of the Case Study Projects

<table>
<thead>
<tr>
<th>Risk Items</th>
<th>Risk Ratings</th>
<th>Risk Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>Project B</td>
<td></td>
</tr>
<tr>
<td>Environmental Risks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-Tech. Technology and certification process uncertainties.</td>
<td>3-4</td>
<td>3</td>
</tr>
<tr>
<td>E-Coord. Stakeholder diversity leading to conflict and misunderstandings.</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>E-SoS. Systems-of-systems environment making control more difficult.</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Risks of using agile methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-Scale. Scalability and criticality of the product.</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A-YAGNI. Use of simple design that does not scale up.</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A-Churn. Personnel turnover with loss of expert knowledge.</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>A-Skill. Not enough people skilled in agile methods.</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Risks of using plan-driven methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P-Change. Rapid change</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>P-Speed. Need for rapid results</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>P-Emerge. Emergent requirements</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>P-Plan. Unrealistic planning, high planning uncertainty.</td>
<td>3-4</td>
<td>2-3</td>
</tr>
<tr>
<td>P-Skill. Not enough people skilled in plan-driven methods.</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Risk rating scale: 0: Minimal risk; 1: Moderate risk; 2: Serious but manageable risk; 3: Very serious but manageable risk; 4: Show stopper risk.

Results Ex-Post Ratings extracted from a Workshop including PM and QM experts from academia and industry


Institute of Software Technology and Interactive Systems
Lessons Learned & Benefits

Lessons Learned of Applying the Hybrid Approach

- **Software delivery** was effective to fulfill contracts with customers and provide competitive products to the market within the planned effort and time plan.
- A systematic, goal-oriented approach for priority setting mitigates the risk of jumping between ideas and not achieving overall goals.
- Agile approaches need a strong framework for success in practice.
- Well-defined milestones can avoid losing the overall perspective on progress goals; the progress of sprint WPs has to be translated to the progress of plan-driven WPs.
- PM planning and control was effective and considerably more efficient than planned.

Benefits from Integrating Agile Sprints in plan-driven PM:

- Improvement of cost, effort, and progress controlling in all parts of the project.
- Transparent overview on needs and status of work for all project participants enabled a very effective and flexible work culture.
- An efficient and tool-supported continuous integration and test process provides visibility of progress and ensures the required software product quality.
- A feature network that provides planning data enables goal-oriented negotiation of the development strategy.
Summary & Future Work

Summary

- The SME company logi.cals has systematically developed a hybrid PM approach for software research and development projects.
- Major innovations in the approach are:
  - Parallel coordinated sprints of software development, research, and marketing.
  - Integrated and very efficient overview on all WPs in the hybrid PM due to a well-integrated tool set, customized to hybrid PM needs and methods.

Future Work

- Evaluation of the hybrid PM approach in research and development groups at a variety of research organizations and SMEs.
- Support of continuous integration and test in engineering environments across organization boarders.
Thank you ...

An SME Transition from Plan-Driven to Hybrid Project Management with Agile Software Development Methods

Stefan Biffl¹, Richard Mordinyi¹, Günter Raidl², Heinrich Steininger², Dietmar Winkler¹

¹TU Vienna, Institute of Software Technology, CDL-Flex, Austria
²logi.cals GmbH, St. Pölten, Austria

Dietmar.Winkler@tuwien.ac.at