Royce' Methodology

Software Engineering I
Lecture 20

Bernd Bruegge
Applied Software Engineering
Technische Universitaet Muenchen

Royce's Methodology

- Demonstration-based approach
  - Identify performance issues and assess intermediate artifacts.
- Architecture-first approach
  - Focus on critical use cases, architecture decisions, and life-cycle plans before committing resources. Address architecture and plan together
- Iterative life-cycle process
  - Each iteration should focus on a specific risk and move the requirements, architecture, and plan in a balanced manner
- Component-based development
  - Minimize human generated lines of code. Use commercial components.
- Change management environment
  - Automate processes to deal with changes introduced by iterations.
- Round-trip engineering
  - Couple models and source code, decreasing cost of change
- Objective quality control
  - Use metrics and quality indicators to assess progress
- Visual modeling languages.
  - Use visual languages to support modeling and documentation.

How much Planning? (Royce)

- The project plan is developed iteratively like the software
  - The plan is refined as the stakeholders increase their knowledge in the application and solution domain
- Planning errors are treated like software defects
  - Early fixing means less impact on project success.
- WBS is organized around software life cycle activities
  - The first level elements in the WBS represent workflows (i.e., management, requirements, design,…).
  - The second level elements represent phases (i.e., inception, elaboration, construction, and transition).
  - The third level elements correspond to artifacts produced during the phases.
- Estimation:
  - Compute the initial estimate with a model
  - Refine it with the project manager, developers, and testers
- After each iteration, revise plan and estimate to reflect the performance of the project and to address planning errors.

How much Reuse? (Royce)

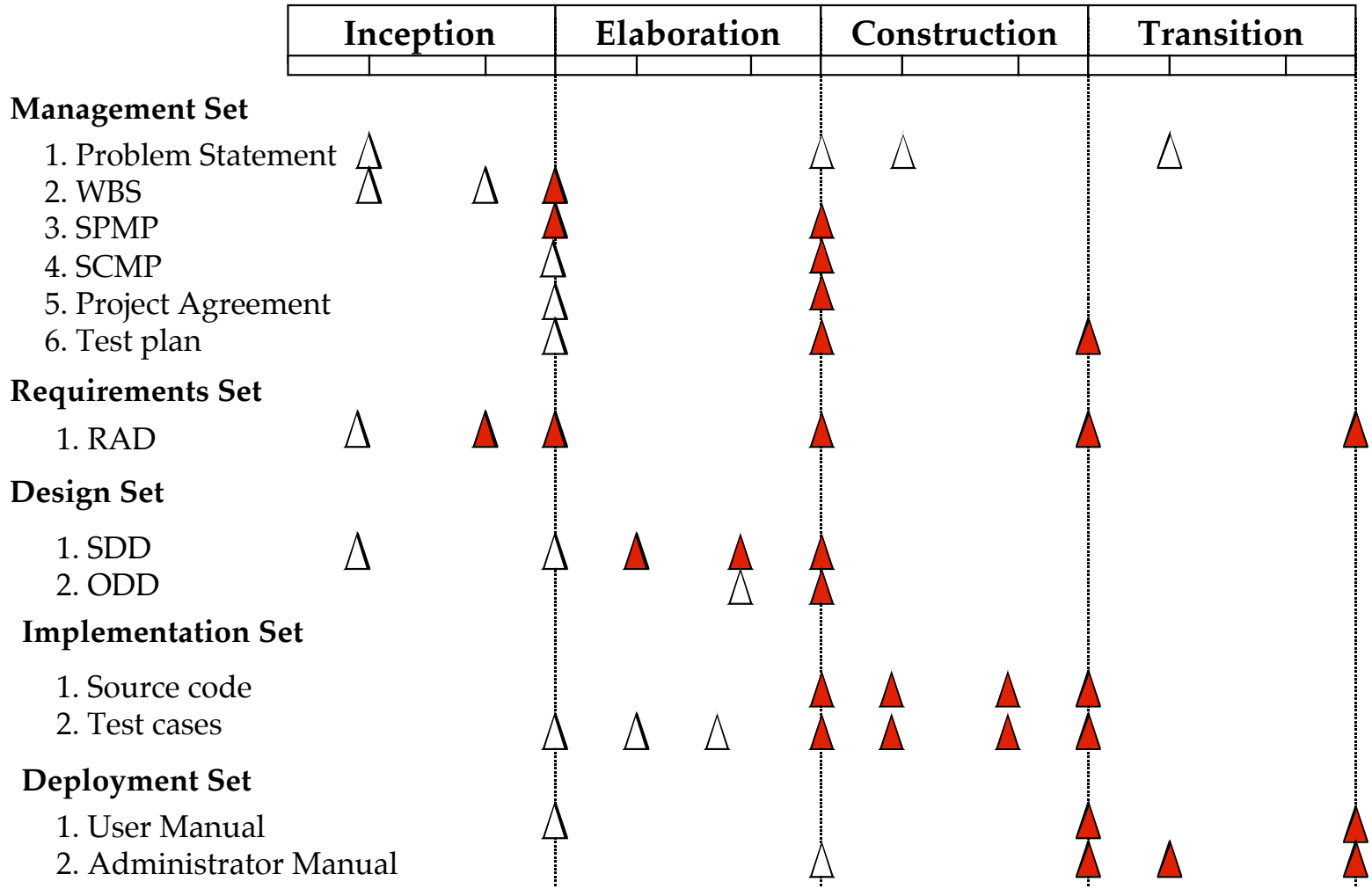
- Buy versus build decisions are treated as risks that should be confronted early in the life cycle (e.g., in the first iterations of the elaboration phase).
  - When components are reused in more than one project, the return on investment can be further increased.
- Key priniciple: Minimize the amount of human-generated source code
  - Reuse commercial components
  - use code generation tools
  - Use high-level visual and programming languages.
- Reuse is treated as a return on investment decision which decreases development time.
  - Mature components and tools also reduce time to repair defects
  - Immature components and tools increase quality problems drastically to off-set any economic benefit.

How much Modeling? (Royce)

- Modeling artifacts based on the activities of the Unified Process
  - Management Set:
    - Artifacts associated with planning and monitoring activities
    - Ad hoc notations to capture the "contracts" among project participants and other stakeholders
    - Problem statement, SPMP, SCMP and status descriptions
  - Requirements set
    - Visionary scenarios, prototypes for user interfaces, requirements analysis model.
  - Design set
    - Software architecture and interface specifications
  - Implementation set
    - Source code, components, executables
  - Deployment set
    - Deliverables negotiated between project manager and client
    - Executable, user manual and administrator manual
- Test artifacts are part of each of the above sets.

Artifact Road Map (Royce)

The diagram of all artifacts sets generated over the phases of a software system



How much Process? (Royce)

- Scale (Most important factor in determining the process)
  - Smaller Projects (1-10 participants)
    - Require much less management overhead
    - Performance depends on technical skills of participant and on tools
    - Focus on technical artifacts, few milestones, no formal processes
  - Larger Projects (more than 10 participants)
    - Management skills of team leaders becomes primary performance bottleneck
    - Well-defined milestones, focus on change management artifacts
- Stakeholder cohesion
  - Cooperating set of stakeholders: flexible plan, informal agreements
  - Contention among stakeholders: formal agreements, well-defined processes
- Process flexibility
  - Rigor of the process definition impacted by rigor of contract
- Process maturity
  - Organizations with mature processes are easier to manage
- Architectural risk
  - Demonstrate feasibility of the architecture before full-scale commitment
- Domain experience
  - Domain expertise shorten the earlier phases of the life cycle.

How much Control? (Royce)

3 management metrics and 4 quality metrics:

- Management metrics:
  - Work. How many tasks have been completed compared to the plan?
  - Costs. How many resources have been consumed compared to the budget?
  - Team dynamics. How many participants leave the project prematurely and how many new participants are added?
- Quality metrics:
  - Change. How many change requests are issued over time?
  - Breakage. How much source code is reworked per change?
  - Rework. How much effort is needed to implement a change?
  - Mean time between failures. How many defects are discovered per hours of testing?

Summary of Royce's Methodology

Issues	Methods
Planning	Evolutionary WBS Initial model-based estimation of cost and schedule (COCOMO II) Iteration planning, including all stakeholders
Modeling	Critical use cases and driving requirements first Architecture first, UML, Round-trip engineering
Reuse	Buy vs. build decisions during elaboration. Focus on mature components
Process	Scale, Stakeholder cohesion, Process flexibility, Process maturity, Architectural risk, Domain experience
Control	Management indicators (work, cost, team dynamics) Quality indicators (change traffic, breakage, rework, MTBF)

References

Summary

Backup and Additional Slides