

# Web Engineering

## Web Application Development Process and Project Management for Web applications

Dec 14 2009

Katharina Siorpaes

### Overview

- Web Application Development Process
  - Selecting a Process
  - Rational Unified Process (RUP)
  - Extreme Programming (XP)
  - Meta-processes
- Project Management For Web Applications
  - Challenges
  - Managing the Development Team
  - Managing the Development Process
  - Project Risks & Risk Management
- Wrap-up

# WEB APPLICATION DEVELOPMENT PROCESS

## The Need for a Formal Process

- Many projects are done “quick and dirty”
  - Pro: shorter development times
  - Con: low quality, i.e. higher operation & maintenance costs
- Two solutions:
  - Adapt existing conventional software process models
    - Models often provide flexibility
    - But are they always a good fit?
  - Develop new Web specific process models

## Definitions: Model vs Method



- Process Model – describe the development approach in the overall context
  - *when* something should be done
  - under *organizational aspects*
    - Examples: RUP, XP
  - Heavyweight vs. lightweight – the degree of process formalization
- Method – describe the development approach in details
  - *how* something should be done
  - *when* it can be done
  - under *content-specific aspects*
    - Example: A UML diagram

## Definitions: Model vs Method



- E.g. If the use of a specific UML diagram is recommended to achieve a specific goal, as practiced in RUP, then this is part of the methods, because the goal pursued here is content specific.
- If it's suggested to program in pairs, as practiced in XP, then this is also a methodological recommendation, since the benefits of such an approach are primarily an improvement in code quality.
- While the decision to use programming in pairs is part of the XP process.

## Definitions: Iteration and Phases



- Iteration – a set of distinct activities that results in a software release
  - Reuse accumulated knowledge
  - The same steps may occur several times
  - Experienced teams, new application domain
- Phase – the span of time between 2 milestones
  - Goal-oriented
  - Risk-oriented
- Literature often erroneously speaks of phases to mean methodological activities, namely requirements definition, analysis, design, implementation, testing, and maintenance. This corresponds to an approach according to the traditional waterfall model, where methodological activities follow one another linearly
- -> Risk handling postponed

## Process Requirements for the Web



- Handling short development cycles
  - Expected duration: 3-6 months
  - Fast-cycle technology & marketing
- Handling changing requirements
  - Many requirements emerge after development begins.
  - Restructuring of data.
  - Evolving technologies & standards.
  - Strong customer involvement.
- Business level decision impacts on requirements for processes.

- Fixed Deadlines vs Flexible Contents
  - “Disposable” releases to demonstrate functionality
  - Time is most critical (very short, e.g. 2-15 days)
- Parallel Development of Releases
  - Small teams working on different versions of the application concurrently
  - Largely a project management problem
  - Emphasis on communication
- Handling quick reactive changes requires prototyping
- Release is defined by date not by feature anymore!
- Requirements become flexible.
- Such a technique is also supported by the fact that metrics for cost in web applications are hard to be used for long term plans.

- Reuse and Integration
  - Coordination among the different projects that will reuse the component
  - Modeling promotes reuse
  - Push the problem towards integration and increase the risk of problem spread across different projects
- Adapting to Complexity Level
  - Process should adapt as development becomes more complex
  - The more complex an application, the more formalized the process should be
- Reuse pushed by short delivery times
- Integration is a bigger problem when involves third parties.

# THE RATIONAL UNIFIED PROCESS

## Rational Unified Process (RUP)

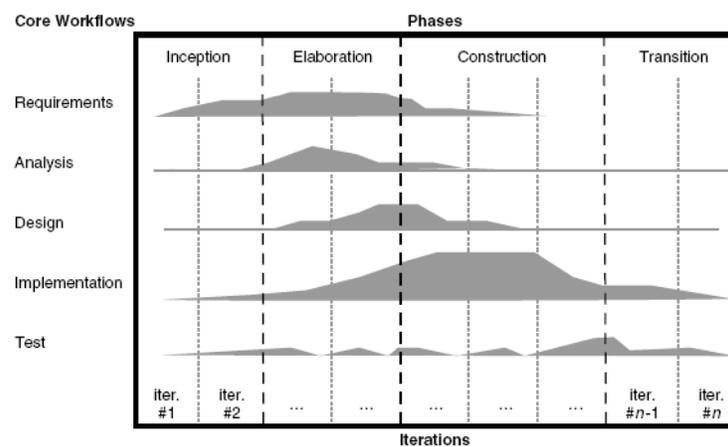
- RUP is a heavyweight process framework.
  - Phase-oriented
  - Incremental
  - Iterative
- Designed for high-complexity, high-quality applications
- RUP methods are grouped into core workflows (or “disciplines”)

## 4 RUP Phases



1. Inception
  - Requirements, Scope, and initial Architecture
2. Elaboration
  - Define architecture, platform, fixed price
3. Construction
  - Finish analysis; design & coding
4. Transition
  - Deliver application to customer

## RUP Core Workflows



## Key Principles Behind RUP



- Adapt the process
- Balance stakeholder priorities
- Collaboration across teams
- Demonstrate value iteratively
- Encourage abstraction
- Focus continuously on quality

## RUP's Suitability for Web Apps



- Inception – POOR
  - assumptions may change as the project progresses
- Elaboration – POOR
  - Developing suitable system outweighs measuring price
  - Internet largely defines system architecture
- Construction - GOOD
- Transition – GOOD
  - in some cases easier because distribution is automatic
  - Overkill (too heavyweight for web apps)
- Fine grained architecture: web tech are changing fast; hard to have a methodology to couple with this.

## RUP & Process Requirements of Web apps



- Short development cycles: POOR
- Changing requirements: POOR
- Fixed deadlines, Flexible content: POOR
- Parallel development: FAIR
- Reuse and integration
  - GOOD reuse
  - POOR integration
- Adapting to flexibility level: GOOD



## EXTREME PROGRAMMING

## Extreme Programming (XP)



- XP is one of the most popular forms of agile processes.
  - Iterative, “test-first”
  - More human-centric/feedback-oriented
- Core Values
  - Communication
  - Simplicity
  - Feedback
  - Respect
  - Courage

## Simplicity



- We will do what is needed and asked for, but no more. This will maximize the value created for the investment made to date. We will take small simple steps to our goal and mitigate failures as they happen. We will create something we are proud of and maintain it long term for reasonable costs.

## Communication



- Everyone is part of the team and we communicate face to face daily. We will work together on everything from requirements to code. We will create the best solution to our problem that we can together.

## Feedback



- We will take every iteration commitment seriously by delivering working software. We demonstrate our software early and often then listen carefully and make any changes needed. We will talk about the project and adapt our process to it, not the other way around.

## Respect



- Everyone gives and feels the respect they deserve as a valued team member. Everyone contributes value even if it's simply enthusiasm. Developers respect the expertise of the customers and vice versa. Management respects our right to accept responsibility and receive authority over our own work.

## Courage

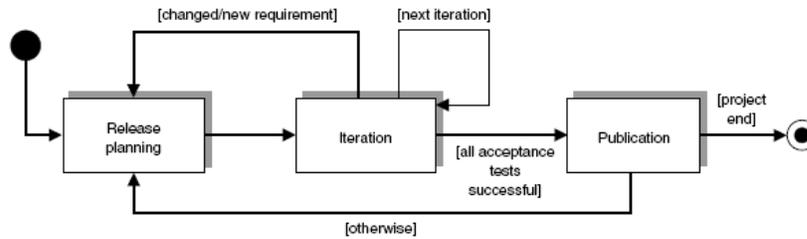


- We will tell the truth about progress and estimates. We don't document excuses for failure because we plan to succeed. We don't fear anything because no one ever works alone. We will adapt to changes when ever they happen.

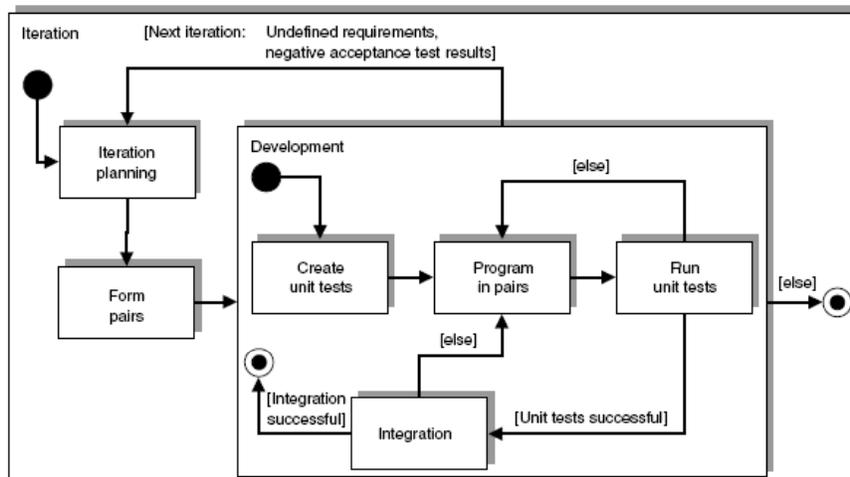
## XP – A Process Overview



- Rapid Successive Releases



## XP – An Iteration View



## XP & Process Requirements

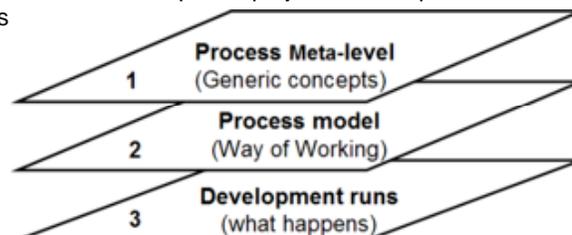


- *Short development cycles*: GOOD fit
- *Changing requirements*: GOOD fit
- *Fixed deadlines, Flexible content*: GOOD fit
- *Parallel development*: GOOD fit
- *Reuse and integration*: POOR fit
- *Adapting to complexity level*: POOR fit

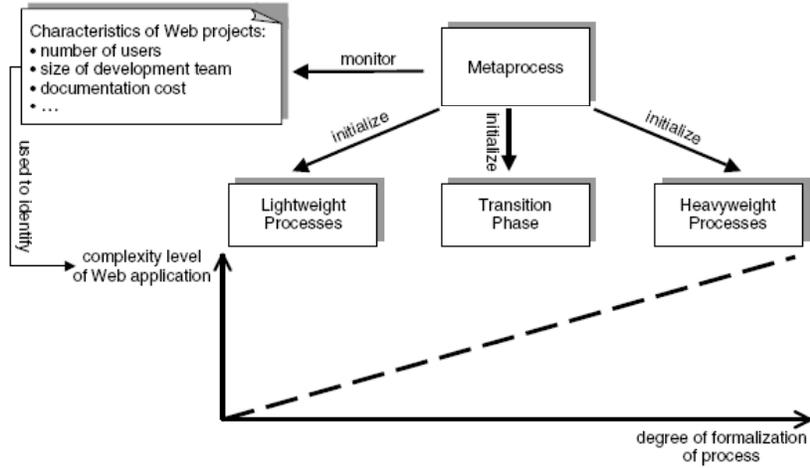
## The Meta-Process Alternative



- Agile processes are generally preferred for the Web, but have 2 main obstacles:
  - Scalability & Complexity
  - High demands on team members
- Handling scalability and complexity occurs over the course of several projects.
- A *meta-process* across software development projects can help manage these changes



## The Meta-Process Alternative



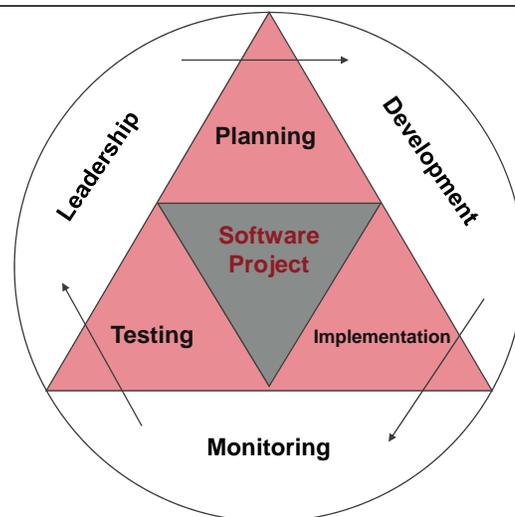
# PROJECT MANAGEMENT FOR WEB APPLICATIONS

## Purpose



- Project management: A systematic approach to planning and guiding project processes from beginning to end.
- It is a human-centered activity
- Like requirements analysis, conflict resolution is critical.
- Many development teams are still “new” to the Web
  - Short history, inexperienced in management
  - Experienced in traditional software only

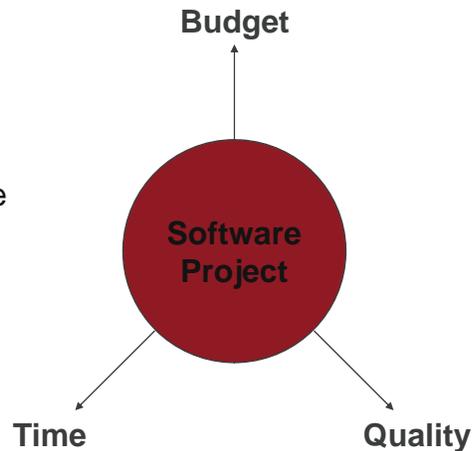
## Objectives & Tasks



## Balancing Conflicting Goals



- A change in one goal incurs trade-offs with the others.
- Be sure all stakeholders are aware of this relationship!



## Traditional vs. Web



- In Traditional Projects:
  - Quality product at lowest cost
  - 10-100 team members
  - 12-18 month horizon
  - \$ millions
  - Requirements-based; structured phases; document-driven
  - OO methods
  - Rigid processes
  - Complex; poor reusability
  - Experienced, professional developers
- In Web Projects:
  - Usable product in shortest time
  - 3-10 team members
  - 3-6 month horizon
  - \$ thousands
  - Agile methods; prototypes
  - Component-based methods; multimedia; visual programming
  - Ad-hoc processes
  - Standardized; high reusability
  - Multimedia designers; Web programmers; marketers

## General Challenges



- Leadership
  - Poor/incomplete planning
  - Unique/legacy software systems
  - Highly technical leadership
- Development
  - Individuality
  - Many alternative solutions
  - Rapid change
- Monitoring

## Development Challenges



- Novelty – unknown & uninformed audiences.
- Usability – no manuals, time to learn extremely low, intuitive!
- Dynamics – time pressures
- Parallelism – subteams & communication
- Continuity – development to transition
- Juvenility – youth; less experience
- Immaturity – inadequate tools

## Product-Related Challenges



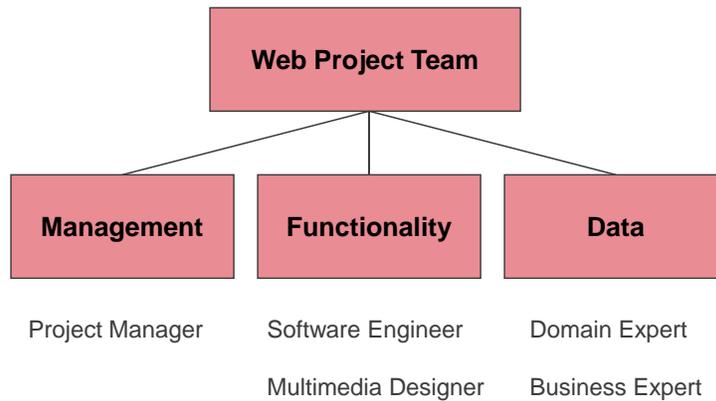
- Web apps are “simple”
- Aesthetics
- Spontaneity
- Ubiquity
- Compatibility
- Stability & Security
- Scalability

## Managing the Development Team



- Success is largely determined by group dynamics & how well they are managed
  - Communication among team members
  - Motivation & coordination by Project Manager
  - Identify & resolve conflicts ASAP
  - Concurrent engineering
  - Primary & backup
  - Documentation is everyone's responsibility

## The Web Project Team



## The Roles of the Project Manager



- Mediator
- Motivator
- Communicator
- Translator
- Trainer
- Customer liaison
  - Requirements during development
  - Post-deployment?

## 10 Golden Rules



1. Promote the professional self-conception of each team member and ethical behavior.
2. Stress the importance of different app knowledge
3. Solve conflicts quickly
4. Keep roles well-defined
5. Look for parallel developments & synergies
6. Spread the documentation task fairly
7. Promote & coordinate continuous use of tools
8. Translate costs & values
9. Keep the customer involved
10. Monitor project progress and objectives

## Managing the Development Process



- Tool-driven vs. document-driven
  - Requirements & test suites
  - Communication (Wikis)
- Configuration management
  - Versioning
  - Short iteration cycles
  - No project is too small for it!
- Measuring progress
  - System specification
  - The application itself

## Project Risks



- Risk: the probability of an activity to result in a loss.
- Most critical risks:
  - Personnel deficits
  - Unrealistic time and cost specs
  - Incompatible third-party components
  - Misunderstood properties
  - Poor user interface

## Nielsen's Top Risks



- Unclear definition of objectives
- Wrong target audience
- Development-oriented page structure
- Inconsistent design
- Insufficient budget for maintenance
- Content recycling & poor linking
- Mixing Internet & intranet
- Marketing research is seen as usability research

- How likely is a problem to occur, what will be the impact, and what are the solutions?
- Assessment: Identify, analyze, & prioritize
- Control: Provision, monitor, mitigate
- Groups are better at assessing and managing risk than individuals.
- Perform a cost-benefit analysis to justify risk management activities.

## WRAP-UP

## Things to keep in mind (or summary)



- A good development process is important
  - Reduce costs
  - Allow to achieve goals
  - Adapts to new problems
- Project Management is part of the meta-development process (process about the process)
  - Minimize risks
  - Enable development process monitoring
  - Requires integration with the development process (“probe” points)

## Bibliography



- Mandatory reading
  - Chapter 9
  - Chapter 10

## Questions?

