

Agile vs. Traditional

Dror Feitelson

Basic Seminar on Software Engineering
Hebrew University
2009

Basic Approach

- Development driven by time constraints
- May sacrifice requirements in order to keep schedule
- Steer project as you go based on user prioritization
- Sustainable effort (limit of 40-hour weeks)

- Development driven by requirements
- May sacrifice budget or schedule in order to fulfill requirements
- Predict schedule and budget in advance and commit in contract
- Deadlines leading to heroic efforts

Process Formalism

- Rapid release cycles
- Iterations are time-boxed
- Scrum: daily meetings for updates and setting the agenda
- Emergent design

- Planned development process
- Phases and iterations with milestones and defined functional success criteria
- Models and documentation

Changes

- “Embrace change”
- Changes are inevitable
- accommodating changes is required to succeed
- Exploit this to also employ new technologies

- “Feature creep”
- Changes are harmful and should be eliminated
- Requirements must be frozen to succeed
- Risks being obsolete on delivery

Iterations and Increments

- Iterative and incremental
- Iterations of days to weeks
- Increments lead to releases to users

- Iterative and incremental
- Iterations of weeks to months
- Increments used for internal structuring and risk management

User Involvement

- Users integrated into team
- Users use each release in real setting and provide feedback
- Continued development based on user priorities

- Acknowledgement of importance of user involvement
- User involvement mainly in requirements elicitation

Testing and Reviewing

- Test driven development: tests are written before code
- Complete system always passes all tests
- All code is reviewed in real time as it is being developed (pair programming)

- Testing workflow done in parallel with other activities
- Developers responsible for unit test
- “Integration hell”
- Code reviews are one of many techniques used