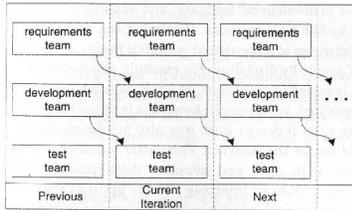


Practice Tips – Project Management

- **Overlapping or pipelining activities across iterations**
 - During iteration N one or more people are doing requirements analysis



Métodos Ágiles para el Desarrollo de Software

1

Prof. Claudia A. Marcos - ISISTAN

Practice Tips - Project Mangement

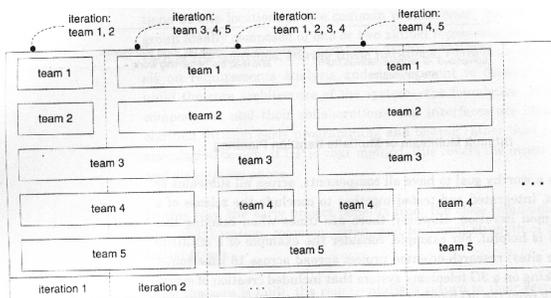
- **Difficult multiteam or multisite common iteration end date**
 - Takes as a rule, at least two of the five teams had to define a common short iteration with integration and no team could go “too long” without participating in a joint iteration
 - Preferred is that all teams work with a common iteration end date

Métodos Ágiles para el Desarrollo de Software

2

Prof. Claudia A. Marcos - ISISTAN

Practice Tips - Project Management



Métodos Ágiles para el Desarrollo de Software

3

Prof. Claudia A. Marcos - ISISTAN

Practice Tips - Project Mangement

- **Multiteam or multisite early development**
 - Early iteration at one location with one or two representatives
 - Emphasis on requirements analysis and development to discover and build the core architecture of the system

Métodos Ágiles para el Desarrollo de Software

4

Prof. Claudia A. Marcos - ISISTAN

Practice Tips - Planning

- Planning meeting ideally held with all developers and customers present or at least include representatives from each team
- During each iteration planning meeting generates and reuse tasks
- In the budget do no forget overhead tasks, demos, management, meetings etc

Practice Tips - Planning

- During each iteration planning, people estimate their total ideal work hours. A common average is around five hours per day
- Promote volunteering for tasks rather than task assignment by a manager when the iteration tasks are defined
- Where possible show all iteration tasks on a wall or whiteboard

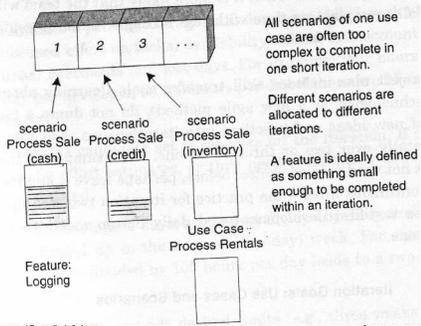
Practice Tips – Iteration Goal

- Iteration length chosen first
 - Most IID methods recommend in the 1-6 week length
 - Smaller teams (such as 10 people) tend towards 1-3 weeks; larger projects with sub-teams (such as 200 people) need more time (4-6 weeks) because of the overhead
 - Early iterations with high rates of discovery and change need more time after stable iterations can be shorter
- The steps
 1. Decide the length of the next timeboxed iterations
 2. Estimate total developer ideal work hour for the iterations
 3. Choose a high-priority request (feature, use case, defect, and so on)
 4. Repeat step 3 until all available resources for the iteration are consumed

Practice Tips – Iteration Goal

- Some use cases are too complex to implement all scenarios (alternative course) within one reasonably short iteration. Perhaps it would take three months to complete all scenarios. So, decomposed it by scenario for ranking, scheduling and tracking. In this case, an iteration should complete the scenarios that it starts; a scenario should not be split across iterations but a use case can be

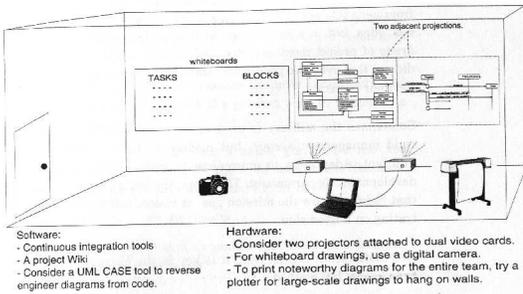
Practice Tips – Iteration Goal



Practice Tips – Environment

- In an agile-oriented project a CASE tool is used for reverse-engineering feature, to support visualization, stock them on walls and to discuss design decisions
- Agile methods promote developments in a common room rather than separate offices, to increase communication
- Agile methods promotes the use of whiteboard space and freehand drawing for creative modeling work
- A digital camera can capture hand-sketched whiteboard model or lists. Then they can print and put the draws on the wall

Practice Tips – Environment



Practice Tips – Requirements

- Iteration by iteration the requirements are defined
- People write during five or ten minutes. Then, they stop and pass their computer to their neighbour, while they themselves receive another computer
- The user is involved in the requirements definition

Practice Tips – Test

- There are automated test for everything
- Tests are written before the code to be tested
- Tests simply pass or fail
- Once the code is written the test is run. If it fails, the programmer debugs until it passes
- As the projects continuous grows all test grow constantly, becoming part of the continuous integration test process, re-executed on each build cycle

Agile Methods Classification

- **Ceremony:** Amount of method weight in terms of documentations, formal, steps, review etc
- **Cycles:** Number and length of iterations. For example, single-pass waterfall has no iterations

Agile Methods Classification

Example: Scrum iterations are exactly 30 days, but is silent on how much or how little ceremony a particular project needs.

