



Software development in practice according to
the project manager and the developer

Tobias Rasmusson & Fredrik Ljungberg

Background

- Who are we?
 - Tobias: Lead developer
 - Fredrik: Project manager
- Purpose of the lecture:
 - Give two perspectives on central aspects of software development
- Focus:
 - Software development projects that produce software applications for users
- Structure of the lecture
 - We have picked seven central aspects of software development
 - We will introduce the aspect and then describe our experiences of it from a developer and a project manager point of view

Aspects of software development

1. Systems development methods (XP, Scrum, RUP, Waterfall, etc.)
2. Time plans and estimations
3. User involvement
4. Quality (control)
5. Requirements management and bugs handling
6. Managing 3rd parties
7. Reflections and suggestions to the students



1 Systems development methods (XP, Scrum, RUP, Waterfall, etc.)

Developer on (1) Systems development methods

- When working as a consultant the customer decides what development method to use.
- A small team (sub project) may use whichever method they like.
- Iterations are fine as long as they meet the fixed deadline

Project manager on

(1) Systems development methods

- Method bibles exist but are rarely used
- The individual project decides processes, systems and documents
- We agree problems with the waterfall model but project deliveries more common but often imply... waterfall!
- Fashion but no magic cure
- Processes important but this is difficult to understand for many people
- People tend to not want not follow release processes, e.g., fix bug AND continue BUT not follow the process
- Convince the customer we must follow the processes

- What is important in the end: Be clear
 - Areas of responsibility
 - Goals/deliverables (what and when)
 - Basic processes

DIADRÒM
DIVDBOW

2 Time plans and estimations

Project manager on 2 Time plans and estimations

- Business value → business case → cost → plans and estimation
- Changing scope is common and a main reason why plans fail
 - Angry customer: "You should have understood that X is included in the scope"
 - Breaking the chain of command: "Person X said Y should be included as well"
- Consequence:
 - A moving target, at what should you aim?
 - You will never make the customer happy
- Clear requirements and processes important for meeting plans and estimations
- Delayed input will imply delayed output
- Time boxes if time is very important
 - Very clear scope, do as much as you can within the time available

Developer on (2) Time plans and estimations

- It is very difficult to estimate how much time it takes to finish a task
- It's alright to be behind schedule – but the deadline is fixed
- Development time may get squashed in a release cycle

DIADRÒM
DIVDBOW

3 User involvement

Developer on (3) User involvement

- Users always enjoy an application prototype
- Users are not always suited to specify requirements, but they must review them
- Some requirements are absurd

Project manager on

3 User involvement

1. Users tend to have good ideas
2. When you start developing a new system, the user suddenly starts to love the old system
3. If you have listened to the users and made that explicit, then you have a useful hostage for the roll out
 - I have listened to you, now you have to support the new system
4. Requirements analysis must be prioritized, rapid feedback is important
5. Deadlines and reviews



4 Quality (control)

Project manager on 4 Quality (control)

- Quality requires processes that are followed
- Quality takes time to achieve
- Automatic testing valuable
- Version 1.0 is usually quite poor quality wise!
- In the end, users give priority to basic qualities; Speed, stability
- There are always known bugs
- Test more than you think
- Acceptance testing

Developer on (4) Quality (control)

- There will be bugs – prepare for them
- Traceability, version numbering and SCM
- Unit testing is great – but make sure someone tests the whole solution.
- Build test applications



5 Requirements management and bugs handling

*Developer on
(5) Requirement management and bug handling*

- Try to take active part in the requirement process
- Know your domain
- Have the PM or customer prioritize bugs and requirements

Project manager on 5 Requirements management and bugs handling

1. Use one dedicated system
 - Excel is rarely good enough
 - Separate lists impossible to manage
2. Integrated systems: Requirements, use case, testing, test planning, bugs handling, etc.
3. Processes for how to prioritize what bugs to handle to a release
 - Do not break the chain of command
4. Communicating with the customer important



6 Managing 3rd parties

Developer on (6) Managing 3rd parties

- Make it clear to everyone that if a component you are dependent on is delayed, then so are your delivery
- The customer may prefer/demand a specific 3rd party
- Try to create a virtual environment for your components

Project manager on 6 Managing 3rd parties

1. Be very clear and write it down
 - What and when to deliver
 - Quantify requirements

2. Acceptance test is key
 - How the delivery will be tested

3. Requirements regarding maintainability
 - Don't forget that someone should maintain the delivery

DIADRÒM
DIVDBOW

7 Reflections and suggestions to the students

Project manager on 7 Reflections and suggestions to the students

- Learn how to program first
- Processes are important
- Girls: You can do it better than you think!
- Boys: You can do it but not as good as you think!

Developer

(7) Reflections and suggestions to the students

- Learn more than one programming language
- Learn how to write scripts
- Write test applications for all new features
- Create the perfect project