Workshops at the start of the project
Proposal for approach
Motivation
Motivation

- We develop software of any size, in our own responsibility up to 15 man-years development scope, and as team members even in significantly larger projects.
  - Since 1992 as a partnership, since 2000 as a public limited company.
- We have seen so many software projects fail and have taken many lessons learned experiences with many customers with us and adapted them into our own processes.
  - Failure is also: not all goals achieved, budget exceeded, schedule exceeded
- We would like to share this knowledge and approach the project "right" or according to best practice.
Pre-existing conditions & scope of Project

- We work with code that is partially 15 to 30 years old
  - It's to be expected that what we're building here could be in use for 15-30 years.

- We are annoyed about the inadequacies of the old software
  - Design errors such as small integers ⇒ Overflowing order numbers
  - Design errors such as tables per plant ⇒ Each task must be adapted x-fold for each plant
  - Software errors as always recurring software errors ⇒ demotivates the employees ⇒ an error must be tested again and again and corrected several times.

- We need to build the project from the ground up
  - Not "Better 10 hours programmed than 1 hour thought" ⇒ Design things properly everywhere and gather all the knowledge of specialist processes and software engineering in workshops
  - Do not skimp on (automated!) testing ⇒ Unit testing and test-driven development ensure high quality, stability, efficiency and user satisfaction
Workshops

Goal: To bring together internal and external knowledge, experience and best practice approaches.

Suggestions for workshops that we consider meaningful

1. coordination of workflow
2. selecting one Toolkit
3. Unit Testing / Test-Driven-Development
4. Techniques
5. Database-Design
6. UI- & UX-Design
7. teaching Scrum, and especially writing user stories
Workshop workflow (I)

- According to which process framework do we work?
  - Suggestion: Scrum according to standard
  - Why do we think it makes sense in this project?
- Training of employees in the process framework
- Who fills which role?
  - Stakeholder ⇒ Specialist user of the customer
  - Product Owner ⇒ "Project manager" on the part of the customer
  - Scrum Master ⇒ K&K Zeljka Parad (Certified Scrum Master)
  - Dev Team ⇒ K&K A-Team, if necessary plus further developers
Workshop Work Process (II)

- Which meeting are held per 2-week sprint?
  - Planning 1 ⇒ Which user stories are implemented in the Sprint? Presentation of user stories and acceptance criteria.
  - Planning 2 ⇒ How are user stories technically implemented?
  - Daily Scrum ⇒ Who is currently working on what? Are there questions, problems, etc?
  - Review ⇒ Demo of all user stories and all acceptance criteria.
  - Backlog Refinement ⇒ Writing, refining and (re-)prioritizing user stories
  - Reflection ⇒ What went good and what went bad in the last sprint? What obstacles are there and how can they be remedied?
Workshop Toolchain

- Where are the user stories maintained?
- Where is the source code maintained?
- Where does the build take place?
- Where are the automated tests performed?
- Where is the Definition of Done maintained?
- Where is the source code documented?
- etc.

Proposal K&K: Microsoft TFS (Team-Foundation-Server) at 
https://tfs.kk-software.de
“Only” Unit Testing or Test-Driven-Development (TDD)?

Do not skimp on automated testing!

TDD is a method of software development, in which a test is written first and only then to implement the software itself.

With TDD every acceptance criteria and every bug gets a Unit Test

- Once a bug is fixed, it will not occur again
- A specification of behavior defined once as a test will never be broken accidentally
  - It is immediately noticeable if conflicting requirements occur
    - All test cases are documented in code and with code
    - I can rethink the new specification or convert the old specification (documented) into a new specification
Workshop Automated Tests (II)
With the focus on Test-Driven-Development

- Development costs and duration are higher in the beginning
  - For each function, an additional (at least!) automated test must be developed
- Software quality will be on a completely different level
  - Much higher user acceptance
- Reducing long-term costs
  - No regressions
  - No manual tests every release or updated necessary
- Discuss in detail at a workshop
Workshop Technology

● How should the project be implemented technically in detail?
● Which code conventions?
● Which IDE?
● How to test (automatically)?
● Which core libraries?
● Which sources (and licenses) for modules?
Workshop Datenbank Design

- Not “Better 10 hours programmed than 1 hour thought” ⇒ Design things properly everywhere and gather all the knowledge of specialist processes and software engineering in workshops.
- The database is one of the most important components. It has the strongest and most long-term impact on the project.
- Here we should clearly justify the design decisions in writing with the considerations (advantages and disadvantages) and make them comprehensible.
Workshop UI- & UX-Design

● Optional: Involvement of an external full-time UX-Designer
  ○ UI = User Interface: What does the program look like?
  ○ UX = User Experience: How does the program feel like?
● Yes or no?
● advantages
  ○ Increased user acceptance
  ○ Uniform user guidance
Workshop Scrum & Writing User Stories

- How do the meetings work in detail?
- How do you work with the Microsoft Team Foundation Server?
- How (and why) are Epic, Feature, User Story, Acceptance Criteria, Team Rules, Definition of Done, and the various Backlogs (Product Backlog, Impediment Backlog) maintained?
- Training of customer product owners
- Joint writing of the first user stories
After the workshops:
⇒ Start of the project
Approaches of K&K
Approaches of K&K

How would K&K build the project?

- Presentation “Process” ⇒ 100% Scrum
  - Recommended prior knowledge: Video about Scrum “Agile Product Ownership in a nutshell”
    https://www.youtube.com/watch?v=502ILHjX9EE
- Präsentation “C# Techniques” ⇒ Coding Guidelines, Design Patterns & TFS
Any questions? Please contact us!

We are committed to your request and will be happy to answer detailed questions promptly. We look forward to your inquiry.

I am your contact person:

Laura Köpl
Head of Marketing and Sales

Phone: 09382 / 3102-241
Mail: koepl@kk-software.de
Acting future-proof
Your K&K Software AG as a reliable partner.