

Workshops zum Projektstart

Vorschlag zum Vorgehen



Motivation



Motivation

- Wir entwickeln Software in jeder Größenordnung, in eigener Verantwortung bis 15 Mannjahre Entwicklungsumfang, als Teammitglieder auch in erheblich größeren Projekten.
 - Seit 1992 als Personengesellschaft, seit 2000 als Aktiengesellschaft
- Wir haben so viele Softwareprojekte scheitern sehen und bei vielen Kunden sehr viele Lessons-Learned-Erfahrungen mitgenommen und in die eigenen Prozesse adaptiert.
 - Scheitern ist auch: Nicht alle Ziele erreicht, Budget überschritten, Zeitplan überschritten
- Dieses Wissen möchten wir gerne teilen und das Projekt gleich “richtig” bzw. nach Best-Practice angehen.

Rahmenbedingungen & Dimension des Projekts

- Wir arbeiten mit Code, der teilweise 15-30 Jahre alt ist
 - Es ist zu erwarten, dass das was wir hier bauen auch 15-30 Jahre in Benutzung sein könnte
- Wir ärgern uns über Unzulänglichkeiten der alten Software
 - Designfehler wie zu kurz gedachte Variablen ⇒ Überlaufende Auftragsnummern
 - Designfehler wie Tabellen pro Werk ⇒ Jede Aufgabe muss x-fach für jedes Werk angepasst werden
 - Softwarefehler wie immer wieder auftauchende Softwarefehler ⇒ demotiviert die Mitarbeiter ⇒ ein Fehler muss immer wieder getestet und mehrfach behoben werden
- Wir müssen das Projekt von Grund auf ordentlich aufbauen
 - Nicht “Lieber 10 Stunden programmiert als 1 Stunde nachgedacht” ⇒ Überall ordentlich konzipieren und mit Workshops alles Wissen von Fachprozessen und Softwareengineering zusammentragen
 - Nicht am (automatisierten!) Testen sparen ⇒ Unit Testing und Test-Driven-Development sorgen für hohe Qualität, Stabilität, Effizienz und Benutzerzufriedenheit

Workshops



Workshops



Ziel: Internes und Externes Wissen, Erfahrungen und Best-Practice-Ansätze zusammenbringen

Vorschläge für Workshops, die K&K für sinnvoll hält

1. Abstimmung **Arbeitsprozess**
2. Abstimmung **Toolchain**
3. Abstimmung **Unit Testing / Test-Driven-Development**
4. Abstimmung **Technik**
5. Abstimmung **Datenbank-Design**
6. Abstimmung **UI- & UX-Design**
7. Abstimmung & Schulung **Scrum** und insbesondere **User Stories schreiben**

Workshop Arbeitsprozess (I)

- Nach welchem Prozess-Framework arbeiten wir?
 - Vorschlag: Scrum gemäß Standard
 - Warum halten wir es bei dem Projekt für sinnvoll?
- Schulung der Mitarbeiter im Prozess-Framework
- Wer hat welche Rolle im Projekt?
 - Stakeholder ⇒ Fachanwender des Kunden
 - Product-Owner ⇒ “Projektleiter” seitens des Kunden
 - Scrum-Master ⇒ K&K Theresa Baumann
 - Dev-Team ⇒ K&K A-Team, ggfs. zzgl. weitere Entwickler



Workshop Arbeitsprozess (II)

- Welche Meetings werden pro 2-Wochen-Sprint abgehalten
 - Planning 1 ⇒ Welche User Stories werden in dem Sprint umgesetzt? Vorstellung der User Stories und Akzeptanzkriterien.
 - Planning 2 ⇒ Wie werden die User Stories technisch umgesetzt?
 - Daily Scrum ⇒ Wer arbeitet gerade woran? Gibt es Nachfragen, Probleme, etc?
 - Review ⇒ Demonstration alle User Stories mit allen Akzeptanzkriterien.
 - Backlog-Refinement ⇒ Schreiben, Verfeinern und (Neu-) Priorisieren der User Stories
 - Reflection ⇒ Was ist im letzten Sprint gut und schlecht gelaufen, welche Hindernisse gibt es und wie kann man sie abstellen?



Workshop Toolchain

- Wo werden die User Stories gepflegt?
- Wo wird der Quellcode gepflegt?
- Wo findet der Build statt?
- Wo werden die automatisierten Tests durchgeführt?
- Wo wird die Definition-of-Done gepflegt?
- Wo wird der Quellcode dokumentiert?
- usw.

Vorschlag K&K: Alles im Microsoft Azure DevOps Server unter <https://tfs.kk-software.de>

Workshop Automatisierte Tests (I)

Was ist uns die Softwarequalität wert?

- “Nur” Unit Testing oder Test-Driven-Development (TDD)?
- Nicht am automatisierten Testen sparen!
- TDD ist ein Vorgehensmodell zur Softwareentwicklung, welches vorgibt, **zuerst einen Test** zu schreiben und erst **danach die Implementierung**
- Beim TDD bekommt jedes Akzeptanzkriterium und jeder Bug einen Unit Test
 - Ein einmal behobener Bug wird nicht wieder auftreten
 - Eine einmal im Test festgelegte Spezifikation oder Verhalten wird niemals *gebrochen* werden
 - Es fällt sofort auf, wenn widersprüchliche Anforderungen auftreten
 - Alle Testfälle sind im Code und mit Code dokumentiert
 - Ich kann die neue Spezifikation überdenken oder die alte Spezifikation (dokumentiert) in eine neue Spezifikation überführen

Workshop Automatisierte Tests (II)

Mit dem Fokus auf Test-Driven-Development

- Entwicklungskosten und -dauer ist am Anfang höher
 - Für jede Funktion muss zusätzlich (mind.!) ein automatisierter Test entwickelt werden
- Softwarequalität wird auf einem komplett anderen Niveau sein
 - Viel höhere Akzeptanz der Benutzer
- Langfristige Kosten senken
 - Bugs, die auftreten werden nach dem Fix mit einem Unit Test abgedeckt, dass dieser Fehler nie wieder auftreten kann
 - Ich muss nicht bei jedem Release oder Update alles manuell testen, sondern es wird automatisch getestet
- Ausführlich an einem Workshop diskutieren

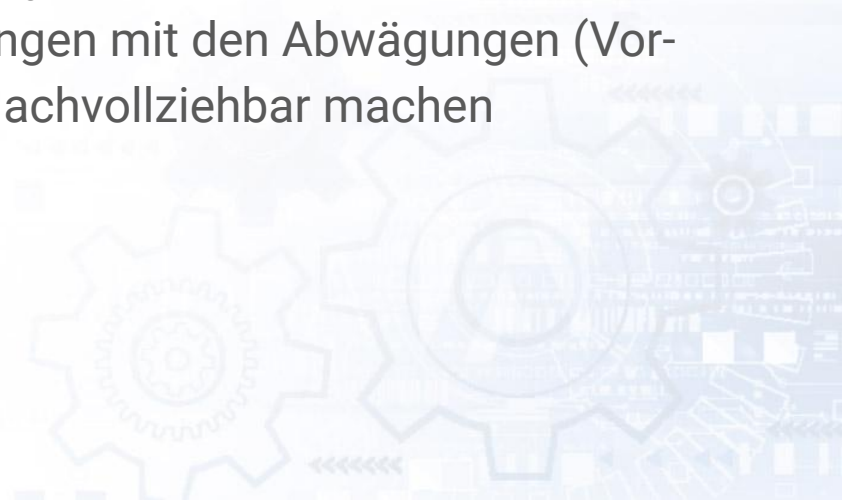
Workshop Technik

- Wie soll das Projekt im Detail technisch umgesetzt werden?
- Welche Coding-Regeln?
- Welche IDE?
- Wie wird (automatisch) getestet?
- Welche Kernbibliotheken?
- Welche Quellen (und Lizenzen) für Module?



Workshop Datenbank Design

- Nicht “Lieber 10 Stunden programmiert als 1 Stunde nachgedacht” ⇒ Überall ordentlich konzipieren und mit Workshops alles Wissen von Fachprozessen und Softwareengineering zusammentragen
- Die Datenbank ist eine der wichtigsten Komponenten, welche die stärksten und langfristigen Auswirkungen auf das Projekt hat
- Hier sollten wir klar die Design-Entscheidungen mit den Abwägungen (Vor- und Nachteile) schriftlich begründen und nachvollziehbar machen



Workshop UI- & UX-Design

- Optional: Hinzuziehen eines externen Vollzeit UX-Designers
 - UI = User-Interface: *Wie sieht das Programm aus?*
 - UX = User-Experience: *Wie fühlt sich das Programm an?*
- Ja oder Nein?
- Vorteile
 - Höhere Benutzerakzeptanz
 - Einheitliche Linie in der Benutzerführung



Workshop Scrum & User Stories schreiben



- Wie laufen die Meetings im Detail ab?
- Wie wird konkret mit dem Microsoft Azure DevOps Server gearbeitet?
- Wie (und warum) werden Epic, Feature, User Story, Akzeptanzkriterien, Team Regeln, Definition of Done, und die verschiedenen Backlogs (Product Backlog, Impediment Backlog) gepflegt?
- Schulung der Kunden-Product-Owner
- Gemeinsames Schreiben der ersten User Stories

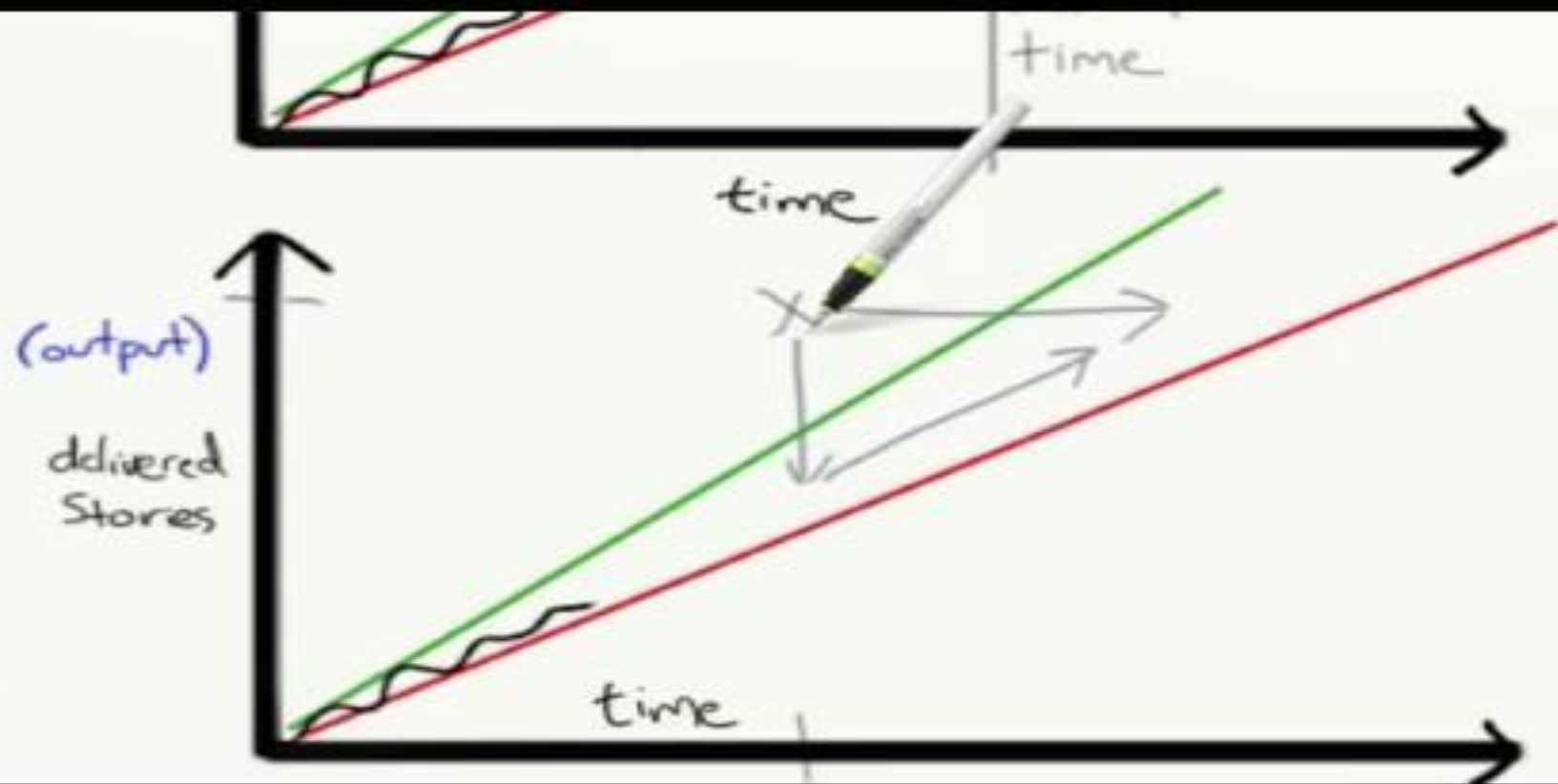


Nach den Workshops:
⇒ Start in das Projekt



Ansätze von K&K

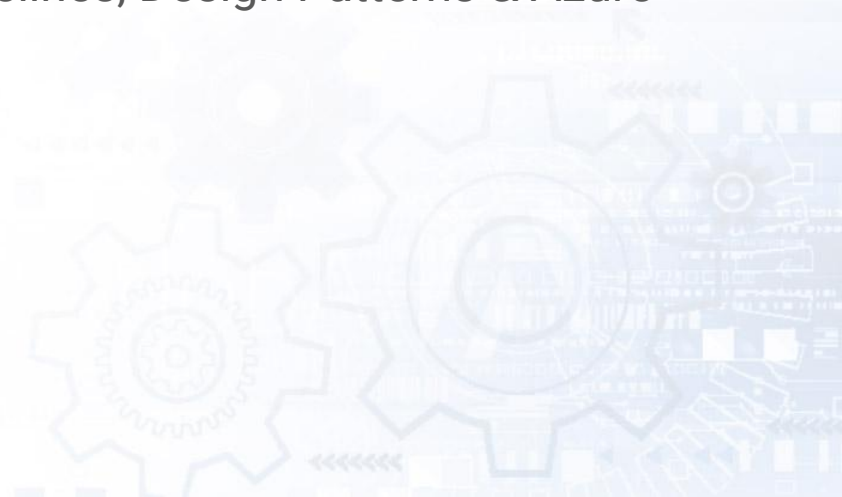




Ansätze von K&K

Wie würde K&K das Projekt aufbauen?

- Präsentation “**Prozess**” ⇒ 100% Scrum
 - Empfohlenes Vorwissen: Video über Scrum “Agile Product Ownership in a nutshell - auf Deutsch” <https://youtu.be/ZUwo8tKoYnQ>
- Präsentation “**C# Technik**” ⇒ Coding Guidelines, Design Patterns & Azure DevOps Server



Fragen? Bitte sprechen Sie uns an!



Wir freuen uns auf Ihren Anruf und Ihre Nachricht. Wir sind um Ihr Anliegen bemüht und stehen für detaillierte Fragen gerne und prompt zur Verfügung.

Ich bin Ihre Ansprechpartnerin:

Laura Köpl

Leitung Marketing und Vertrieb

Telefon: 09382 / 3102-241

E-Mail: koepl@kk-software.de





Zukunftssicher agieren

Ihre K&K Software AG als verlässlicher Partner.