



TRANSITION
TECHNOLOGIES

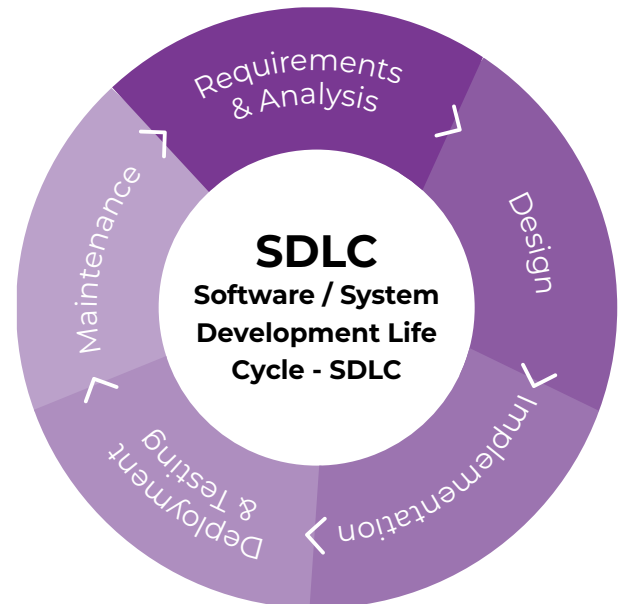
Wie wir unsere Software-SDLC Methodik entwickeln

SLDC-Methodiken

– Einführung zum SDLC

Um Ihnen die Möglichkeit zu geben, zu verstehen, wie wir Software entwickeln, lassen Sie uns mit einer kurzen Einführung in den Software Development Lifecycle (SDLC) beginnen.

SDLC ist ein Skelett eines Softwareentwicklungsprozesses, der aus verschiedenen Phasen besteht, die beteiligt sein sollten. Im weiteren Verlauf erfahren Sie mehr über unsere Herangehensweise an jede dieser Phasen, aber lassen Sie uns einen kurzen Blick auf SDLC-Methodologien werfen, die den Entwicklungsprozess ergänzen.



– SLDC-Methodiken

Introduction

Mit den oben genannten Bausteinen könnte man sagen, dass sie bereits einen Prozess bilden, dem man folgen kann. Die Phasen könnten einfach nacheinander ausgeführt werden, gerade nach Abschluss der vorherigen Phase. So ist es auch. Die Methodik heißt Wasserfall.

Bei TTPSC sind wir jedoch der Meinung, dass der einfachste Ansatz nicht immer der beste ist. Nachdem wir über hundert Projekte geliefert haben, haben wir genug herausgefunden, um zu sagen, dass es andere Methoden gibt, die die Softwareentwicklung viel besser erleichtern. Einer unserer Lieblingsmethoden ist Scrum, die von einem agilen Ansatz kommt.

Warum wir uns für Agile entscheiden?

In der modernen Welt können wir uns nur einer Sache sicher sein: Sie verändert sich.

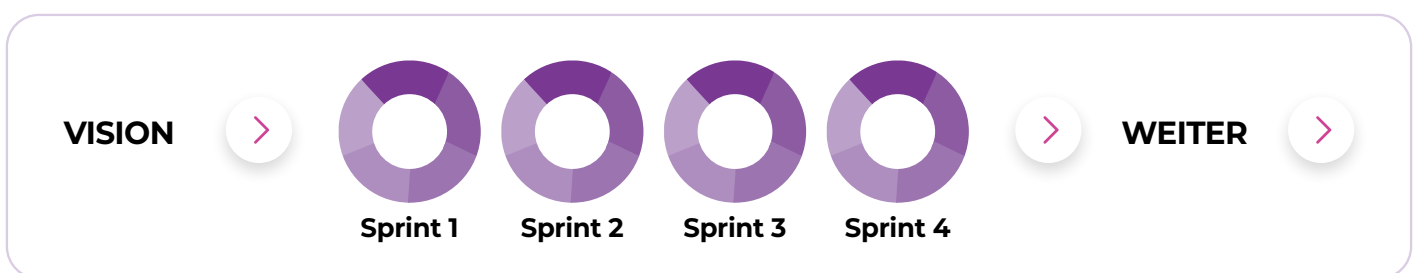
Technologie, Geschäft und sogar die Bedürfnisse unserer Kunden ändern sich schnell. Sehr oft dauern Projekte Monate oder sogar Jahre. Während dieser Zeit kann sich das Umfeld dramatisch verändern - die jüngste Situation im Zusammenhang mit COVID-19 ist ein gutes Beispiel dafür. Beim Wasserfall-Ansatz kann ein Projekt, das in einer sehr frühen Phase entworfen wurde, sein Ziel komplett verfehlen. Nicht weil das Design schlecht war - einfach weil sich das Ziel in der Zwischenzeit geändert hat.

AGILE erlaubt es uns, bei Bedarf die Richtung zu ändern, sehr oft ohne oder mit sehr überschaubaren Auswirkungen auf die Lieferfristen.

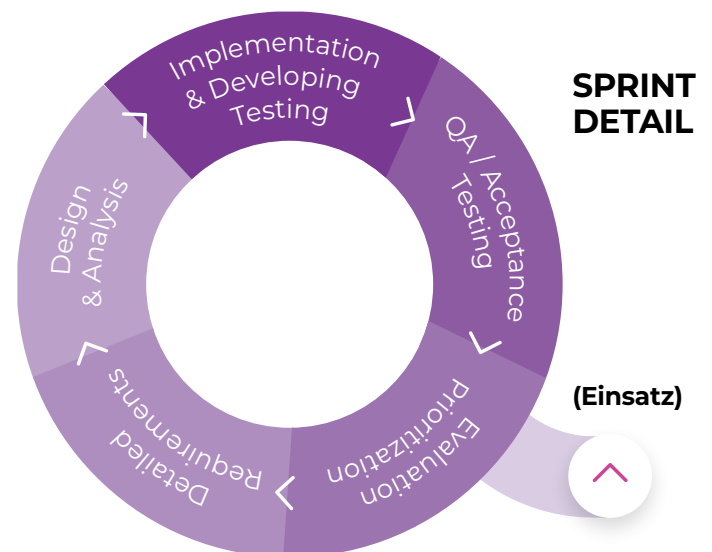
Scrum für Kundenzufriedenheit

Scrum, eine der agilen Methoden, orientiert sich an den Bedürfnissen der Kunden. Die Arbeit wird in so genannten Sprints (Iterationen) realisiert.

Ihnen geht eine Planungszeremonie voraus, die in einem Sprint Backlog mündet, das einen Plan darüber darstellt, was während des Sprints realisiert werden soll. Die Aufteilung der Arbeit in einzelne Sprints ermöglicht die Abschwächung eines der Risiken, da die Entwickler auf die anstehende Arbeit und nicht auf das gesamte Produkt fokussiert sind. Die gesamte Produktvision wird im Product Backlog festgehalten, das für jeden Stakeholder des Produkts transparent ist. Innerhalb eines Sprints arbeitet das Team an einem bestimmten Teil der Funktionalität des Produkts, aber für diesen Teil sind alle Phasen involviert. Das bedeutet, dass nach einem Sprint das gelieferte Produktinkrement entworfen, implementiert, getestet und mit bereits bestehenden Funktionalitäten integriert wurde. Nach dem Sprint wird das Inkrement in einem speziellen Review-Meeting bewertet. Dies gibt dem Kunden die Möglichkeit, das bisher Geleistete zu begutachten und Feedback zu geben. Basierend auf diesem Feedback werden die nächsten Iterationen geplant. Um ständig nach Verbesserungsmöglichkeiten im Prozess zu suchen, wird auch ein Retrospektiv-Meeting abgehalten, bei dem das Team entscheidet, welches Element verbessert werden könnte und plant, wie dies geschehen soll.



Um auf die Bedürfnisse des Kunden zurückzukommen, werden die Sprintlänge, die Zeremonien und Meetings mit anderen Teams (falls es solche gibt) abgestimmt, sowie alle Schritte, die der Prozess des Kunden erfordert, so dass unser Kunde keine zusätzliche Arbeit leisten muss, um das Inkrement des Produkts zu integrieren. Gut implementiertes Scrum erlaubt es Teams, fließend die Richtung zu wechseln, ohne den stabilen Lieferrhythmus zu stören.



Dieser Ansatz kommt auch den Kunden selbst zugute: Sie können in konstanten Zyklen ein Inkrement ihres bestellten Produkts sehen und fundierte Entscheidungen über die nächsten Schritte treffen. Wir hoffen, dass es auch Ihnen gefällt.

So machen wir es in jeder Phase

Anforderungen und Analyse

Die Methode wird von den Bedürfnissen unserer Kunden abhängen.

Einige haben ihre Product Owner, die mit unseren Teams zusammenarbeiten und ihnen gut beschriebene Backlog Items liefern, wir kümmern uns um den Rest. Die anderen haben ihre Erwartungen

an die Software im Detail beschrieben und brauchen nur unsere Product Owner, um die Erwartungen in die Backlog Items zu übersetzen und das Team dazu zu bringen, entsprechend der Spezifikation zu liefern. Schließlich gibt es Kunden, die große Ideen über ein Produkt haben, das sie gerne hätten, aber sie brauchen eine Anleitung, wie sie ihre Vision in ein reales Produkt verwandeln können. Bei solchen Kunden können unsere Product Owner ihr ganzes Können zeigen.

Nach dem ersten Gespräch mit dem Kunden besteht der erste Schritt darin, die Stakeholder zu identifizieren. In der Regel lassen sich diese in zwei Gruppen einteilen: interne und externe. Interne Stakeholder sind in der Regel Personen aus verschiedenen Bereichen des Unternehmens, die Einblicke in das Produkt geben können (z. B. Recht, Compliance usw.) oder davon betroffen sind (z. B. Betrieb, interne Benutzer). Externe Stakeholder sind in der Regel die Endbenutzer des zu erstellenden Produkts.



Wenn der Product Owner die Anforderungen für das zu erstellende Produkt sammelt, muss er daran denken, dass das Produkt zwar für die Endbenutzer bestimmt ist und deren Bedürfnisse und Erwartungen erfüllen muss, aber auch die Ziele der Organisation erfüllen muss.

Deshalb ist es so wichtig, die internen und externen Stakeholder zu identifizieren und mit ihnen zusammenzuarbeiten. Nach den ersten Gesprächen erstellt PO eine Vision des Produkts und teilt sie mit den Stakeholdern. In dieser Phase ist es wichtig, den Kunden zuzuhören und sie mit einzubeziehen, denn ohne ihren Input wird der PO nur Annahmen treffen und keine fundierten Vermutungen anstellen.

Eine der Techniken, die zur Darstellung des Produkts verwendet wird, ist **User Story Mapping**.

Es zeigt die Wege der verschiedenen Benutzertypen durch das Produkt und was sie erreichen können, wenn sie einen bestimmten Weg absolvieren. Nach einer solchen Übung haben der PO, die Stakeholder und das Entwicklungsteam ein gemeinsames Verständnis davon, was gebaut werden soll. Zu diesem Zeitpunkt werden die gesammelten Informationen in der Regel im JIRA-Tool von Atlassian gespeichert, das Transition Technologies PSC üblicherweise für die Verfolgung der Anforderungen und des gesamten Projektstatus verwendet.

Natürlich wird der Product Owner in dieser Phase nicht allein gelassen. Oft ist es so, dass für die tiefgehende Analyse einiger Anforderungen eine Expertise aus einem bestimmten Bereich benötigt wird. Zum Beispiel beschäftigen einige nicht-funktionale Anforderungen in der Regel unsere Experten, wenn es um Sicherheit oder Performance geht. Diese sind oft nicht offensichtlich, denn wenn man über ein Produkt nachdenkt, ist die Funktionalität das, was Kunden in der Regel am meisten interessiert.

Wir sind immer gerne bereit, Sie mit unserer Erfahrung bei der Spezifikation Ihrer Anforderungen zu unterstützen. Denken Sie daran, dass wir in Zyklen arbeiten und sich diese und alle folgenden Phasen ständig wiederholen, bis das Produkt fertig ist.

Design (Architektur & UX/UI)

Die Tatsache, dass wir in einem agilen Ansatz arbeiten, bedeutet nicht, dass wir ohne ein gutes Design mit der Implementierung fortfahren.

Dies bezieht sich sowohl auf das UX/UI-Design, das angibt, wie die Software aussehen und sich verhalten soll, als auch auf das Design der Softwarearchitektur, das die Elemente der Software und die Beziehungen zwischen ihnen beschreibt. Bei der Erstellung des User Interface/Experience Designs beginnen wir mit unseren Kunden bei Null und ermitteln den Hauptzweck der Software. Wir legen die wesentlichen Geschäftsziele fest und setzen Prioritäten, die den Umfang des Projekts definieren.

In der Forschungsphase wollen wir unsere Kunden und Endbenutzer so gut wie möglich kennenlernen. Das Ziel der Forschungsphase ist es, alle qualitativen und/oder quantitativen Daten zu sammeln, die uns die tatsächlichen Bedürfnisse und Hintergründe bewusst machen, die es uns ermöglichen, ein nützliches Produkt zu entwickeln.



Dann gehen wir durch den Strategieteil. Im Vergleich zu anderen ähnlichen Produkten auf dem Markt definieren wir, wie wir dem Produkt einen Mehrwert verleihen wollen und inwieweit es den Bedürfnissen, Problemen und Zielen unserer Kunden entspricht. Durch Modellierung und Prototyping bestimmen wir den Inhalt und Umfang der Anwendung, die Hauptmerkmale und Funktionalitäten, die für den Erfolg des Projekts erforderlich sind. Wir erstellen Low- und High-Fidelity- Mockups, die dem Endprodukt sehr nahe kommen. Unabhängig davon, in welchem Stadium der Entwicklung wir uns befinden, sollten wir unser Produkt so bald wie möglich testen. Diese Phase ist entscheidend, um Feedback zu unserem Inkrement zu sammeln und mögliche Dysfunktionalitäten zu erkennen.

Zu guter Letzt über die User Experience - in Transition Technologies PSC glauben wir, dass Produkte, die wir entwickeln, auch für die Benutzer mit Behinderungen zugänglich sein sollten.

Warum sollte Ihr Produkt z.B. für eine sehbehinderte Person unzugänglich sein?

Wir sehen auch keinen Grund. Deshalb sind unsere Produkte standardmäßig konform mit den Web Content Accessibility Guidelines (WCAG). Als nette Zugabe erfüllen sie die gesetzlichen Vorschriften zur Barrierefreiheit.

Wenn es um Architektur geht, sind wir davon überzeugt, dass eine starke architektonische Kompetenz im Team viel Entwicklungsarbeit einsparen kann und die entstehende Software einfacher zu warten ist. Deshalb versichern wir neben der natürlich erwarteten Erfahrung, dass unsere Architekten in den neuesten Trends gut ausgebildet sind.

Bei der Planung der Architektur beginnen sie mit der Analyse. Sie halten sich nicht an einen einzelnen Ansatz und sind sowohl für eine monolithische als auch für eine Microservices-Architektur offen. In jedem Ansatz denken unsere Spezialisten daran, die Verantwortlichkeiten von Modulen (oder Microservices) aufzuteilen, so dass das Prinzip der einzelnen Verantwortung befolgt wird. Sie erstellen keine Module, die eng gekoppelt sind. Dies ermöglicht den Austausch ganzer Komponenten und macht sie austauschbar, was Ihnen eine große Flexibilität bei Planänderungen gibt.

Während des Architekturdesigns folgen wir Regeln wie DDD (Domain-Driven Design), die es uns ermöglichen, das Problem unseres Kunden tief zu verstehen und es richtig anzugehen. Bei der Entscheidung, welche Technologie zum Einsatz kommt, bewerten wir auch die tatsächlichen Bedürfnisse. Obwohl wir bei TT PSC ein bewährtes Set von Technologien und Frameworks "erster Wahl" entwickelt haben, das uns bei der Entscheidungsfindung hilft. Wir verwenden sie nicht einfach in jedem Fall wieder, sondern suchen nach der besten für eine bestimmte Herausforderung.

Implementierung

Hier passiert die Magie (Codierung). Im Laufe der Jahre haben die Entwickler bei TT PSC eine Menge Erfahrung mit verschiedenen Sprachen, Frameworks und Tools gesammelt.

Das, kombiniert mit aktuellem Wissen über die neuesten Technologien und dem Bewusstsein für Sicherheitsaspekte, machte es uns ziemlich einfach, eine Menge Standardisierung in der Phase einzuführen, in der der Code geschrieben wird.



Unsere Ingenieure legen sehr viel Wert auf die Qualität des Codes, daher haben wir die Regeln, wie der Code geschrieben werden sollte (Coding Standards) für verschiedene Programmiersprachen.

Um sicherzustellen, dass der von uns produzierte Code den Regeln folgt, verwenden wir Tools wie SonarQube für die statische Code-Analyse (es kann sogar noch mehr, da es zum Beispiel potenzielle Schwachstellen im Code anzeigen kann).

Zusätzlich decken wir den Code während der Implementierung mit automatischen Unit-Tests ab. Sie werden verwendet, um zu überprüfen, ob die Ausführung einer Codeeinheit (z. B. einer Methode) unter den erwarteten Bedingungen/Parametern das gewünschte Ergebnis liefert.

Dies ist sehr hilfreich, wenn wir später den bestehenden Code modifizieren - unerwartete Änderungen im Codeverhalten werden automatisch erkannt und der Entwickler wird darüber informiert. Schließlich wird der Code, bevor er Teil des Produkts wird, einem Code-Review durch einen anderen Entwickler unterzogen - was bedeutet, dass mindestens zwei Ingenieure bestätigen, dass er unseren Standards entspricht.

Einsatz & Test

Wir profitieren so weit wie möglich von Continuous Integration/Continuous Deployment-Praktiken und -Tools, um automatisierte Tests in einem frühen Stadium einzuführen.

Dank dessen wird der Code häufig integriert und eingesetzt, was uns erlaubt, jedes Problem (Fehler) schnell zu entdecken und die notwendigen Korrekturen im nächsten Integrationszyklus bereitzustellen. Dies geht Hand in Hand mit der Philosophie der häufigen Überprüfung und Anpassung.

Durch die automatische und regelmäßige Ausführung des Deployments wird das Deployment in die Produktionsumgebung viel einfacher und weniger herausfordernd als es sein könnte.



Sobald der Code implementiert ist, können wir die Qualität weiter verifizieren (da dies ein untrennbarer Bestandteil wertvoller Produkte ist). Testen ist für uns keine separate Phase. Es ist ein Teil der täglichen Arbeit, eingebaut in bereitgestellte Funktionen. Um dies zu erreichen, wartet der fertige Code nicht auf eine Testphase, um Fehler zu finden. Alle Qualitätsprüfungen finden während einer Iteration statt, und sie werden von funktionsübergreifenden Teams durchgeführt.

Das Testen ist eine fortlaufende Aktivität, die von allen Teammitgliedern durchgeführt und verantwortet wird. Daher setzen wir Best Practices wie Unit-Tests, automatisierte Tests, TDD (Test Driven Development) oder BDD (Behaviour Driven Development) usw. ein. Beim Testen dürfen wir die Sicherheitsaspekte nicht vergessen. Dies wird heutzutage extrem wichtig, wenn die Zahl der Cyberangriffe sehr schnell wächst.



Als Antwort darauf haben wir eine Reihe von qualifizierten Spezialisten in diesem Bereich, die den Titel eines Certified Ethical Hackers tragen. Sie sind bei uns, um mögliche Schwachstellen in der Anwendung zu entdecken (und den Entwicklern zu ermöglichen, diese zu entfernen), bevor die Software von außen angegriffen werden kann.

Wartung

Sobald die Software an unsere(n) Kunden ausgeliefert ist, lassen wir sie nicht allein.



TT PSC ist bereit und willens, Ihnen bei der Nutzung Ihres Produktes zu helfen oder es durch Hinzufügen neuer Funktionalitäten weiter zu entwickeln. Wenn Sie nicht mehr brauchen, bekommen Sie natürlich eine Garantie, und unseren Support (auch in verschiedenen Zeitzonen).

Möchten Sie mehr darüber erfahren, wie wir arbeiten? Nehmen Sie einfach kontakt mit uns auf!

**Kontaktieren
Sie uns!**

Transition
Technologies PSC

contact@ttpsc.com
www.ttpsc.com



TRANSITION
TECHNOLOGIES