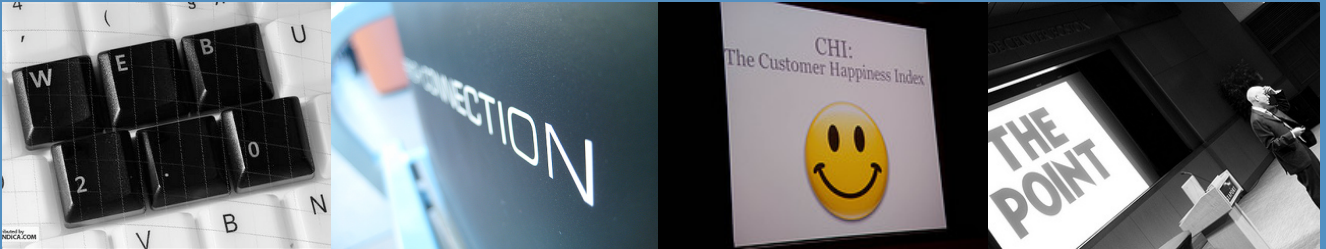


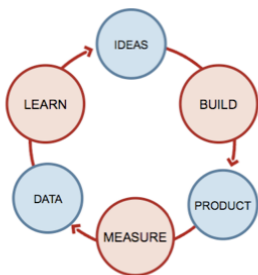
Continuous Delivery



02. Februar 2011

Kontinuierliche Auslieferung hochwertiger Software

„Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.“ Als die Autoren des Agile Manifesto diesen Satz formulierten, zögerten sie damit, diesen als zentrales Prinzip der agilen Softwareentwicklung herauszustellen. Ihre Begründung: Der Satz sei zu trivial. In der Praxis jedoch, stellen wir immer wieder fest, wie sehr dieses „triviale“ Prinzip unterschätzt wird, besonders zum Zeitpunkt der Softwareauslieferung.



Das Problem ist bekannt: Wie kann eine neue Idee schnellstmöglich umgesetzt und an den Kunden ausgeliefert werden?

Ob die Realisierung einer Idee erfolgreich sein wird, ist in den seltensten Fällen absehbar. Erst wenn die Software in die Produktivphase gebracht und echtes Feedback eingeholt wurde, kann eingeschätzt werden, ob sie den Anforderungen des Kunden gerecht wird.

Je mehr Zeit für die Verwirklichung einer guten Idee benötigt wird, desto größer ist die Gefahr, dass die Annahmen, die bei der Konzeption getroffen wurden, in der Produktivphase nicht mehr gültig sind. Als Beispiel für diese Problematik, kann Flickr erwähnt werden. Nach einer aufwändigen Konzeptions- und Entwicklungsphase startete Flickr als Online-Game-Plattform, bis die Ergebnisse einer Untersuchung zeigten, dass sich ihre Anwender viel mehr für Fotosharing und andere Onlineaktivitäten interessiert haben. Das (Flickr) GNE-Projekt¹ musste aus Kostengründen nach zwei Jahren eingestellt werden.

Aber nicht nur das Risiko, welches das späte Einholen von Feedback mit sich bringt, sollte berücksichtigt werden, sondern auch die Gefahr dass die Software oder einige Features überhaupt nicht ausgeliefert werden können.

In der Praxis wird schnell erkannt, dass je seltener eine Software released wird, desto größer ist die

Wahrscheinlichkeit, dass Probleme in der Testphase auftreten werden. Die Release-Phase wird meistens zum Alptraum.

Continuous Delivery geht vom einfachen Prinzip aus, dass die Software jeder Zeit einsatzbereit ist. Mit Hilfe von agilen Ansätzen wie Continuous Integration und Continuous Testing wird versucht, den Releaseprozess so zu gestalten, dass die Software per Knopfdruck ausgeliefert wird. Hierdurch wird nicht nur sichergestellt, dass das mit dem Release verbundene Risiko minimiert wird, sondern auch, dass das Endprodukt inkrementell an die Bedürfnisse des Kunden oder Stakeholder angepasst werden kann.

Die Auslieferung sollte eine Geschäftsentscheidung sein



Wann der Knopf letztendlich gedrückt wird, sollte eine reine Geschäftsentscheidung sein und nicht durch Releasezyklen und Ähnliches bestimmt werden.

Besonders bei strategischen Entscheidungen, verschafft diese Möglichkeit einen erheblichen Vorteil gegenüber Konkurrenten. Wer am schnellsten reagiert überlebt.■

Jackson Santos

Tags: Continuous Integration, Continuous Deployment, Deployment Pipeline, Value Map Stream

¹ Game Neverending (Quelle: Wikipedia/EN)