

Agiles Multi- Projektmanagement

Weniger ist mehr

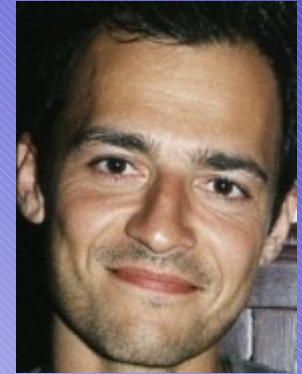
Alexander Kriegisch



<http://scrum-master.de>

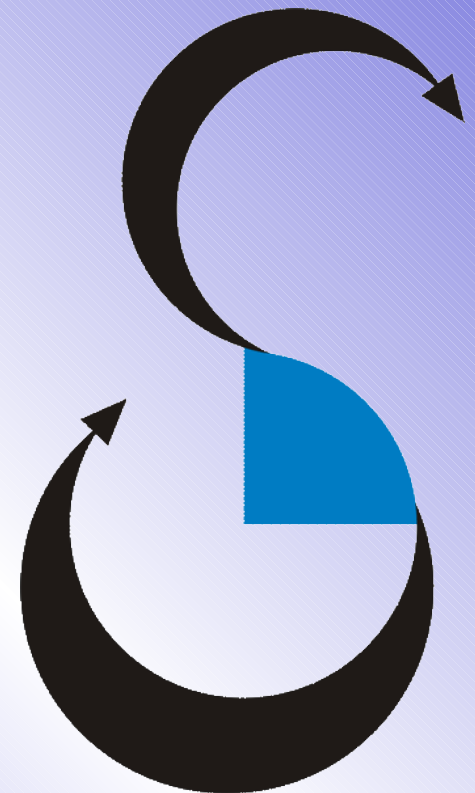
Scrum-Master.de – Agiles Projektmanagement

- Alexander Kriegisch
 - Freiberuflicher agiler Coach & Projektmanager
 - 20 Jahre kommerzielle Programmier-Erfahrung
 - 12 Jahre Vollzeit in Software-Industrie
 - 8 Jahre Beratung (Presales, Partner, Projekte)
 - 5 Jahre Projektmanagement
- Scrum-Master.de
 - Training, Coaching, Supervision
 - Themen: Agile Enterprise, Scrum
 - (Multi-)Projektmanagement
 - Troubleshooting
 - Interim-Management



Inhalt

- **Scrum im Einzelprojekt**
Was macht agile Teams so (hyper-)produktiv?
- **Erfolgsfaktor Priorisierung**
Warum ist Multitasking böse?
- **Agiles Multi-Projektmanagement**
Scrum-Prinzipien projektübergreifend anwenden
- **Engpaßtheorie und kritische Kette**
Was hat die Theory of Constraints mit Scrum zu tun?
- **Butter bei die Fische!**
Multi-Projekt-Szenarien im Simulator
- **Und die Moral von der Geschichte ...**
Zusammenfassung





Scrum im Einzelprojekt

Was macht agile Teams
so (hyper-)produktiv?

Erfolgsfaktoren in Scrum-Projekten

- **iteratives Vorgehen** (Sprints) → Zyklus „apply, inspect, adapt“
- **einfache Regeln** („simple, not easy!“)
- **wenige Rollen** mit klar abgegrenzten Kompetenzen
 - Product Owner → fachliche Sicht
 - Team → technische Sicht
 - Scrum Master → Prozeßsicht
- **Planung** auf Makro- und Mikro-Ebene
 - Product Backlog → Gesamtplanung (Projekt, Release)
 - Sprint Backlog → Iterationsplanung (wenige Wochen)
- **Controlling / Reporting** auf Makro- und Mikro-Ebene
 - Product Burndown → Blick aufs Gesamtprojekt
 - Sprint Burndown → Blick auf die Iteration
 - Tasks → Einzelaufgabe pro Bearbeiter
- **Timeboxing** für Meetings und Sprints → Pünktlichkeit!

Das Wichtigste – last, but not least

- **klare Priorisierung** der Anforderungen durch den Auftraggeber (Product Owner)
 - nach Business Value in absteigender Folge
 - das Wichtigste / Wertvollste zuerst
 - im Notfall bleibt Unwichtiges liegen
- **Vermeidung von Multitasking** bei Einzelaufgaben (Tasks) und Anforderungen / Features
 - nicht alles gleichzeitig anfangen
 - Aufgabe abschließen, bevor die nächste begonnen wird

Diese beiden Punkte werden im Folgenden noch wichtig...



Erfolgsfaktor Priorisierung

Warum ist Multitasking böse?

Multitasking ist geschäftsschädigend!

- Kontextwechsel kosten Zeit
 - Unterbrechung des Arbeitsflusses
 - min. 1h, bis man wieder im „Flow“ ist und effizient arbeitet
- Erhöhtes Risiko durch
 - mehrere offene Baustellen,
 - hohen Bestand unfertiger Erzeugnisse und somit
 - immens hohe Kapitalbindung
- Kumulation von Effekten auf mehreren Ebenen
 - Arbeitsrhythmus des einzelnen Mitarbeiters
 - Bearbeitungsstand eines einzelnen Projekts
 - Koordination bzw. mögliche Konflikte (Termine, Ressourcen) vieler gleichzeitig laufender Projekte → Domino-Effekt



Agiles Multi-Projektmanagement

Scrum-Prinzipien
projektübergreifend anwenden

Status Quo im Multi-Projektmanagement

- Ressourcenverteilung auf viele Projekte gleichzeitig nach Gießkannenprinzip, bestenfalls proportional zu Projektrang
- an allem wird parallel gearbeitet, nichts wird (rechtzeitig) fertig
- Ressourcenkonflikte durch Terminverschiebungen
- massive Kapitalbindung in unfertigen Erzeugnissen
- Multi-Projektmanagement = Multitasking als Kunstform

- ABER: Wenn es **wirklich** wichtig ist, bekommen oftmals Einzelprojekte Fokus und werden konzentriert abgearbeitet, und das funktioniert.
- Frage: Wieso macht man das erst im Eskalationsfall so?

Lösungsansatz „Agile Enterprise“

- Wie kann eine große Organisation wieder agil werden?
 - Overhead eliminieren
 - mehr direkte Kommunikation im internen Kunden-Lieferanten-Verhältnis
 - strengere Priorisierung notwendiger Tätigkeiten auf allen Organisationsebenen (und sich daran halten!): nicht alles gleichzeitig anfangen, sondern Wichtiges zuerst machen
 - Termine realistisch setzen und einhalten, aber sich nicht bzgl. „Iron Triangle“ in die eigene Tasche lügen
 - weniger Multitasking auf allen Ebenen!!!
 - aufhören, sich einzubilden, irgendjemand könne ein Jahr oder länger im voraus exakt planen
 - sich nicht mehr vormachen, Pläne würden sich nicht ändern

Agiles Multi-Projektmanagement

- Idee: Verwaltung des Projekt-Portfolios in einem **Project Backlog**
 - Priorisierung nach Business Value
 - sofortige Realisierung wichtiger Projekte, andere müssen warten
 - Timeboxing (Eintaktung) für Iterationen und Release-Zyklen
 - Ressourcen-Konzentration auf Wichtiges und Wertvolles
- Erzielbare positive Effekte
 - kürzere Projektlaufzeiten (Time to Market, RoI)
 - weniger Ressourcenkonflikte
 - geringere Kapitalbindung
- Notwendige Voraussetzungen
 - Wille des Auftraggebers, konsequent zu priorisieren
 - Top-Management spielt mit



Engpaßtheorie und kritische Kette

Was hat die Theory of Constraints mit
Scrum zu tun?

Engpaßtheorie nach E. M. Goldratt

- Theory of Constraints (ToC) entstammt der Systemtheorie der Kybernetik
 - System besteht aus Einzelementen, die miteinander wechselwirken und klar voneinander abgegrenzt sind
 - System wächst so lange, bis Engpaß auftritt
 - Engpaß-Erkennung erfordert Blick auf Gesamtsystem, da jedes Subsystem, isoliert betrachtet, optimal funktionieren kann
 - Anwendung ursprünglich in den Bereichen Logistik und Fertigungssteuerung
 - Übertragung auf Projektmanagement durch Goldratt
 - Engpaß = kritische Kette (critical chain)
 - PM-Methode = Critical Chain Project Management (CCPM)

Quelle: http://de.wikipedia.org/wiki/Theory_of_Constraints

Engpaß-Identifikation und -Beseitigung

1. Engpaß identifizieren: Das System ist eine Kette, die nur so stark sein kann, wie das schwächste Glied.
2. Engpaß maximal ausnutzen: Wenn das System nur so stark sein kann, wie das schwächste Glied, muß das schwächste Glied maximal ausgenutzt werden.
3. Alles dem Engpaß unterordnen: Prioritäten verschieben, damit der Engpaß wirklich maximal ausgenutzt wird, z.B. eine Maschine, die als Engpaß identifiziert wurde, in Sonderschichten mit mehr Personal laufen lassen.
4. Engpaß verbreitern: Wenn maximale Engpaß-Nutzung nicht genügt, muß durch zusätzliche Maßnahmen der Engpaß auf ein höheres Kapazitätsniveau gehoben werden, z.B. zusätzliche Maschine bestellen oder Engpaß-Team personell aufstocken.
5. Bei Schritt 1 erneut beginnen.

Wenn durch Schritt 4 ein Engpaß behoben wurde, wird das System durch den nächsten Engpaß begrenzt. Daher ist diese Vorgehensweise ständig zu wiederholen.

Quelle: http://de.wikipedia.org/wiki/Theory_of_Constraints

Critical Chain Project Management (CCPM)

- Zeitreserven nicht aktivitätsbezogen, sondern als großer Block am Ende des Projekts
- Grund: menschliche Tendenz, jeglichen eingeplanten Puffer für Aktivitäten aufzubrauchen („Studenten-Syndrom“)
- Netzpläne werden rückwärts vom Projektende her geplant
- kritischer Pfad wird identifiziert
- 50% des gesamten Puffers ans Ende des kritischen Pfads
- restliche 50% des Puffers verwenden, um kritischen Pfad gegen Störungen zu schützen
- Ziel des CCPM: entstandene Kette schützen, Aufgaben frühestmöglich beginnen lassen

Quelle: http://de.wikipedia.org/wiki/Critical_Chain_Management

Parallelen zwischen Engpaßtheorie (ToC) und Scrum

ToC

- klare Priorisierung von Aktivitäten
- Multitasking vermeiden
- Engpässe beseitigen
- ständige Optimierung des Systems

Scrum

- streng priorisiertes Product Backlog
- Multitasking vermeiden
- Hindernisse beseitigen (Scrum Master)
- kontinuierliche Verbesserung des Scrum-Prozesses



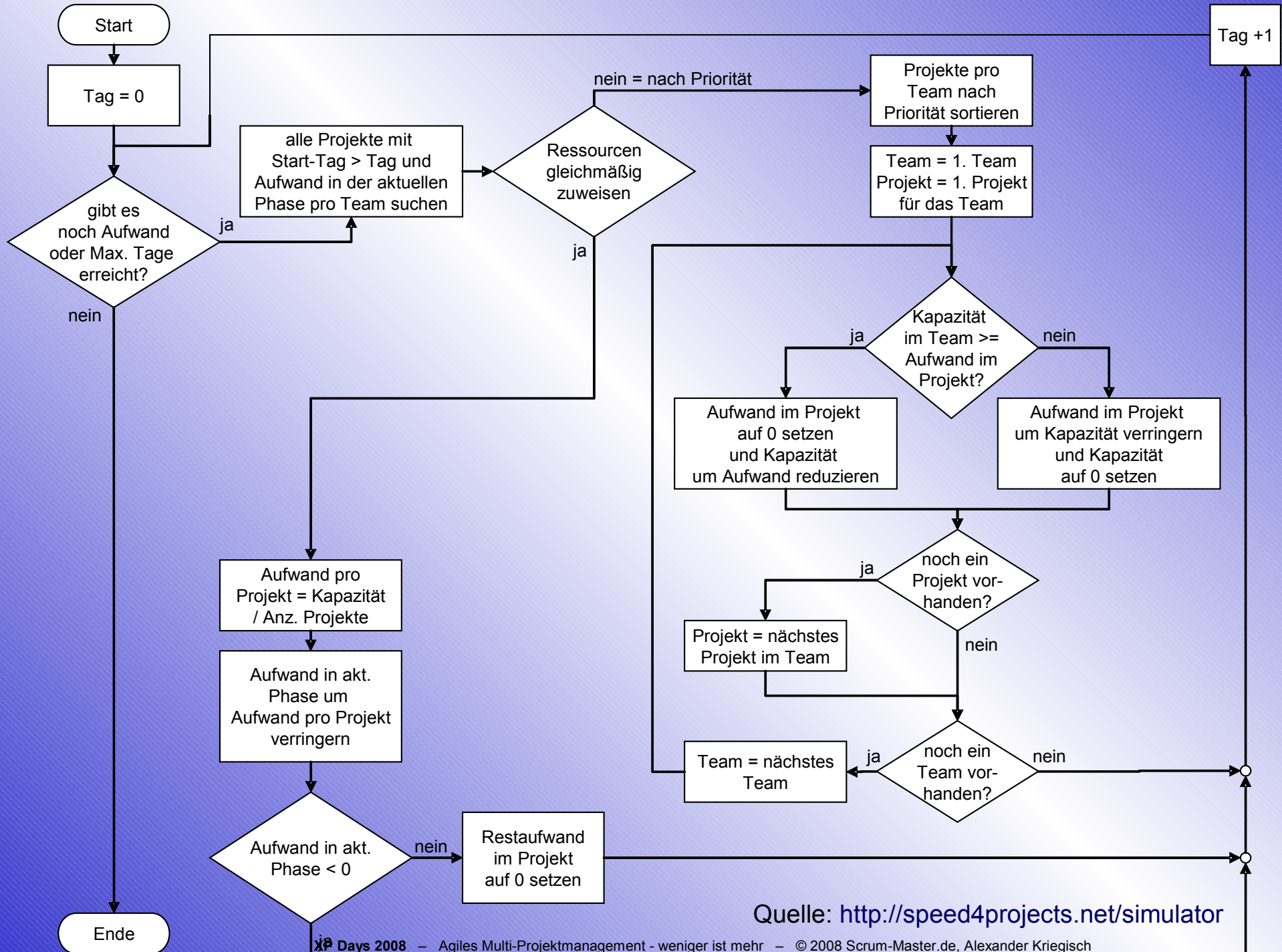
Butter bei die Fische!

Multi-Projekt-Szenarien
im Simulator

Übersicht Multi-Projekt-Simulator

- Werkzeug zur Simulation von Ressourcen-Zuteilungs-Strategien auf mehrere Projekte
- basierend auf MS Excel™ mit Makros in Visual-Basic™
- implementiert einen Zustandsautomaten, welcher in Rechenblättern hinterlegte Multi-Projekt-Szenarien (Projekte, Phasen, Teams, Kapazitäten, Aufwände) für verschiedene Arten der Ressourcenzuteilung durchrechnet
 - gleichverteilt: alle Projekte erhalten abwechselnd Ressourcen
 - first come, first serve: Projektphasen werden in chronologischer Reihenfolge abgearbeitet
 - Priorisierung: frei definierbar, z.B. nach Projektröße, Ertrag/Tag, RoI, Ertrag pro Aufwand im Engpaß (also ToC-basierend)
- Quelle: <http://speed4projects.net/simulator>, initiiert von Wolfram Müller (Projektcontroller bei 1&1 Internet AG)
- Verbessert und um agiles Szenario erweitert durch Alexander Kriegisch (Scrum-Master.de)

Ablaufdiagramm Multi-Projekt-Simulator



Quelle: <http://speed4projects.net/simulator>



Live-Simulation

Multi-Projekt-Szenarien
im Simulator

Live-Simulation

- An dieser Stelle im Vortrag fand eine kurze Simulation statt, welche hier nicht wiedergegeben werden kann.
- Bei Interesse an einer Vorführung oder Erläuterung, z.B. via Web-Konferenz oder Desktop Sharing, wenden Sie sich bitte über das Kontaktformular auf Scrum-Master.de an Herrn Kriegisch.
- Gleiches gilt, wenn Sie die um das agile Szenario und einige kleine Fehlerbehebungen erweiterte Version des Simulators haben möchten.
- Im Folgenden werden zusammenfassend nur die Ergebnisse einiger ausführlicher Simulationsreihen vorgestellt.

Ergebnisse Simulation mit 4 klassischen Projektphasen

- Gleichverteilung der Ressourcen ist die mit Abstand schlechteste Strategie
- Jegliche Art der Priorisierung steigert den Durchsatz (Ertrag pro fertigem Projekt pro Tag) mindestens um 150%, also auf mehr als das Doppelte.
- First come, first serve ist die am wenigsten effektive der getesteten Priorisierungen.
- ToC-gemäße Priorisierung (Ertrag pro Aufwand im Engpaß) ist am effektivsten (>600% Durchsatz-Steigerung).
- Bereits eine geringfügige Aufweitung des Engpasses bringt bei priorisierten Strategien eine nochmalige deutliche Verbesserung, die allerdings bei der ToC-Strategie weniger deutlich ins Gewicht fällt.

Ergebnisse Simulation mit 10 agilen Iterationen

- Wechsel von 4 starren Projektphasen auf 10 Iterationen mit gleichmäßiger Team-Auslastung bringt bei identischen Projektaufwänden nochmals deutliche Durchsatzsteigerungen bei (fast) allen Strategien.
- Gleichverteilung bleibt suboptimalste Strategie, verdreifacht aber trotzdem den Durchsatz durch iteratives Vorgehen.
- First come, first serve und RoI-Priorisierung steigern den Durchsatz immerhin um 73% bzw. 66% gegenüber Wasserfall-Modell mit 4 Phasen.
- ToC-Priorisierung verbessert sich gegenüber Wasserfall-Modell nur um 17%, aber immerhin.
- ToC ist im agilen Szenario immer noch fast 3x so gut wie Gleichverteilung gegenüber 7x im Wasserfall-Szenario.



Und die Moral von der Geschichte

...

Zusammenfassung

Zusammenfassung

- Scrum und Agilität entfalten ihre ganze Kraft nicht im Einzelprojekt auf einer „agilen Insel“, sondern erst im flächendeckenden Multi-Projekt-Einsatz.
- Agile Prinzipien lassen sich auf Project Offices und ganze Unternehmen übertragen.
- Multitasking ist geschäftsschädigend!
- Strenges Priorisieren wird unweigerlich belohnt durch höheren Durchsatz im Einzelprojekt sowie in Multi-Projekt-Umgebungen.
- Es lohnt sich, ein besonderes Augenmerk auf Engpässe zu legen, die Planung an ihnen auszurichten und sie aufzuweiten.

Vielen Dank!

Alexander Kriegisch



<http://scrum-master.de>

The logo for Scrum-Master.de, featuring a stylized 'S' made of two curved arrows, one black and one blue, forming a circle. To the right of the logo, the text "SCRUM-MASTER.DE" is written in a large, bold, black, sans-serif font.