

# Vorstellung Agile Methoden

Marc Schöchlin

[ms-github@256bit.org](mailto:ms-github@256bit.org)

Stuttgart, 22. Juli 2020

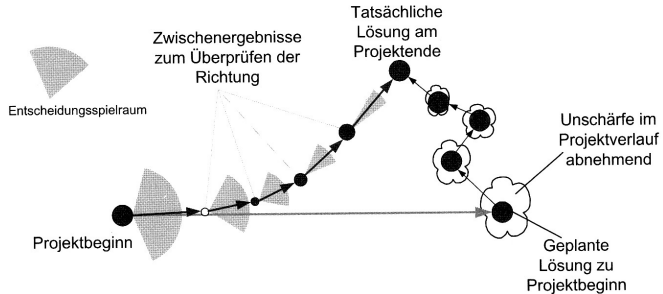
# Überblick

- 1 **Einführung**
  - Der klassische Weg
- 2 Agiles Arbeiten
- 3 Kanban
- 4 Scrum
- 5 Abschluß

# Gründe für Projektunsicherheiten

- Kunde weiss am Anfang nicht genau was er braucht
- Auftragnehmer versteht nicht genau, was der Kunde will
- Auftragnehmer unterschätzt / überschätzt Aufwände (Planung)
- Nicht alle Anforderungen und Stakeholder sind bekannt
- Kunde hat widersprüchliche Anforderungen, z.T. wg. Politik
- Politik, Management Risiken
- Änderungen in den Prioritäten, Geschäftsprozessen etc. während des Projektes
- Projekt in komplexe Projektlandschaft eingebunden
- Technische Risiken, z.B. Infrastruktur hält nicht was sie verspricht

# Projektfortschritt und Entscheidungsspielraum



# Klassische Antworten und Situationen

- Anforderungen
  - Ziele: genaues Verständnis der Anforderungen, Anforderungstabilität, (Verfolgbarkeit)
  - Am Anfang des Projekts möglichst vollständig abstimmen (Bei Risiken in Stufen vorgehen)
  - Möglichst genaue Spezifikation/Pflichtenheft erstellen
  - Änderungen nur über ein striktes Change Request Verfahren
- Architektur
  - Ziele: Architektur = stabiles Gerüst der Software
  - In früher Projektphase entwickeln (mindestens Grobarchitektur)
  - Anforderungen späterer Phasen berücksichtigen
- Dokumente
  - Ziele: Klarheit, Review-Fähigkeit bzw. Abstimmbarkeit, Stabilität und Reduktion von Kommunikation
  - Alles wird dokumentiert (Prinzip der Schriftlichkeit), wichtig sind insbesondere Spezifikationsdokumente und Architekturdokumente
  - Für Offshore-Partner, Wartungsteam, das Team

# Klassische Antworten und Situationen

- Risikomanagement
  - Projekte werden in der Regel in (wenigen) Stufen (3-12 Monate) durchgeführt
  - Aktiver (häufig unsystematischer) Umgang mit Risiken (Einzelmaßnahmen wie technischer Durchstich, Risikoliste, ...)
  - Risiken häufig nur pauschal berücksichtigt
- Pläne
  - Am Anfang des Projektes grobe Planung (auch zur Preisfindung),
  - Feinplanung bei Bedarf
  - Wenn's knapp wird: Weglassen/Verschieben unwichtiger Funktionen oder Qualitätseinbußen (z.B. Testaufwände kürzen)
  - Ziele nach Priorität : (1) Einhaltung des Termins, (1) Einhaltung des Budgets, (3) Qualität, (4) Vollständige Funktionalität
- Verträge
  - Festpreisprojekte zur Kostenkontrolle
  - Preis im Verhältnis zur Funktionalität schwer einschätzbar

# Klassische Antworten und Situationen

## ● Kunde

- integriert in Phasen abhängig von seiner eigenen Verfügbarkeit
- ist über Anforderungsdefinition, Dokument-Reviews und System- und Abnahmetest einbezogen
- Entwicklung: selten gemischte Teams bedingt durch Projektphasen
- Kommunikation zum Teil reglementiert durch Projektleiter („Ein-Ansprechpartner“-Modell) oder über Dokumente
- Ziele: Langfristige Kundenbindung, Vertrauen, verbindliche Absprachen

## ● Prozesse

- Aktivitäten / Workflows im Detail festgelegt
- Viele Rollen
- Viele, genau festgelegte Artefakte (Doku, Code und Co.)
- Ziele: Planbarkeit, Wiederholbarkeit, Verbesserbarkeit (CMM)

# Auswirkungen

- Standard Projekteskalation: Grün ... Gelb ... Dunkelrot ... Gelb
- Projektplan und Realität driftet stark auseinander
- Change Requests machen große Schmerzen (Projektverzug, Kosten)
- Große Teile der Fachlichkeit ergeben sich während des Projekts
- Kunde bekommt eigentlich nicht, was er will, sondern vielleicht noch nicht mal das was im Vertrag steht
- Definierte Prozesse sind oft schwer durchzuhalten
- Projektziel muß mit massivem Aufwand erkaufte werden
- Projektziel muß mit massiver Scopereduktion erkaufte werden
- Chef-Architekten-Denkmäler
- weniger Spaß für Projektmitarbeiter



# Überblick

## 1 Einführung

## 2 **Agiles Arbeiten**

- Allgemein
- Kernaspekte

## 3 Kanban

## 4 Scrum

## 5 Abschluß

# Manifesto for Agile Software Development

Was ist uns wichtiger?

***Individuals and interactions***

over processes and tools

***Working software***

over comprehensive documentation

***Customer collaboration***

over contract negotiation

***Responding to change***

over following a plan

► 2001 - <http://agilemanifesto.org>

# Die agile Vorgehensweise

- Das Team übernimmt Verantwortung
- Ausrichten der Arbeitsweise auf die kontinuierliche Änderung von Anforderungen
- Probleme iterativ lösen, „Minimal viable Product“
- Bewusst sich um Kommunikation bemühen
- Ein kontinuierlicher Ergebnisfluß steht im Fokus
- Produktivität und nicht Effizienz steht im Fokus
- Konzentration auf möglichst nur eine Aufgabe
- Stetiger Wissenstransfer, Wissens-Silos vermeiden
- Arbeiten in Zyklen
- Transparenz mit allen Beteiligten
- Konsequente Umsetzung
- Aktives Arbeiten an der kontinuierlichen Verbesserung
- Fehlerkultur, Respektvoller und lösungsorientierter Umgang mit Fehlern
- Ermitteln von KPIs zur Erfolgsmessung

# Auswirkungen auf die Teamkultur

- Gemeinsame Verantwortung - kooperativer Führungsstil
- Das Team organisiert sich weitgehend selbst
- Führung findet über das Vermitteln von Visionen und Strategie statt
- Teamleiter unterstützt Team die Aufgaben möglichst eigenverantwortlich zu bearbeiten
- Entscheidungen werden hauptsächlich im Team gefällt
- Feedbacks bzw. Verbesserungsprozesse finden regelmäßig und häufig statt (Retro und OneToOne)
- Kontinuierliches Lernen steht im Vordergrund
- Peering, gemeinsames Bearbeiten von Problemen
- Daily Standup Meeting: Was war, was wird, wo gibt es Schwierigkeiten?
- Persönlicher Kontakt mit allen Beteiligten wird aktiv angestrebt
- Projektteams bilden sich abteilungsübergreifend

# Überblick

- 1 Einführung
- 2 Agiles Arbeiten
- 3 Kanban**
  - Einführung
- 4 Scrum
- 5 Abschluß

# Grundsätzliches

## Definition

Kanban is an approach to change management.

It isn't a software development or project management lifecycle or process.

David Anderson / Vater des Software-Kanban (2007)

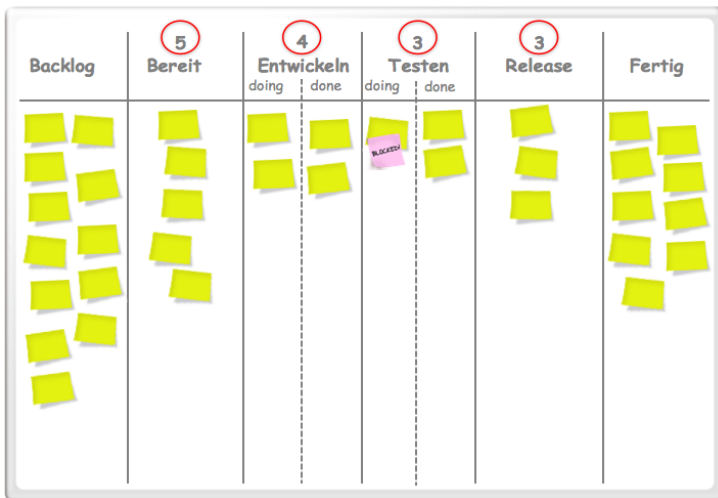
## Kanban in a nutshell

- Small, cross-functional teams
- Product split into small, roughly, estimated stories
- Iterations
- Continuous improvement

# Kanban in der Anwendung

- Gut geeignet für ungeplante Tätigkeiten: Bugs, Optimierungen, Betriebsprobleme, reaktive Themen, Servicemanagement, ...
- Gut geeignet für gut überschaubare Entwicklungsprozesse
- Begrenze die Menge begonnenner Arbeit
- Verkürze die Zeit zwischen Beginn und Erledigung einer Aufgabe
- Konzentration und Fokussierung auf die aktuelle Aufgabe
- Vermeiden von Taskwechsel
- Visualisiere den Fluss der Arbeit
- Mache die Regeln für den Prozess explizit (Publizieren wann wie gehandelt wird)
- Miss und steuere den Fluss

# Kanban Board





# Kanban Board

DevOps Kanban

Schnell-Filter: Nur meine Vorgänge Zuletzt aktualisiert

5 Aufgaben Max. 10 1 OnHold Max. 2 1 Wird Ausgeführt Max. 3 6 Abnahme PO Max. 5 1 Fertig Veröffentlicht...

~ Beschleunigen 2 Vorgänge

~ Alles andere 12 Vorgänge

SUP-349  
↑ Stage Hybris-Server @ eMail-Adressen

SUP-354  
↑ Neuer MA L...

SUP-355  
↑ Typsystem Update am 12.3.2015

SUP-357  
↑ fashion.man nicht eingespielt?

SUP-359  
↑ Checkbilder fehlen

SUP-312  
↑ Oracle Testsystem bei P...

SUP-322  
↑ Zabbix für ... erreichbar machen

SUP-328  
↑ Zugriff für Cukes auf Test 1

SUP-318  
↑ Umlaute im Log auf test und stage

SUP-325  
↑ Datenbankpools Hybris anpassen

SUP-328  
↑ Anpassung hat IP-Adressen-Bereich erweitert. Anpassungen notwendig?

SUP-347  
↑ Existiert unter ... ein Ordner /services?

SUP-358  
Release Hotfix 3.16.1

SUP-360  
Hotfix branch 3.16.1 erstellen

# Überblick

- 1 Einführung
- 2 Agiles Arbeiten
- 3 Kanban
- 4 Scrum**
  - Grundsätzliches
  - Im Detail
  - Auswirkungen
- 5 Abschluß

# Begriff



Scrum (engl. das Gedränge) ist ein Vorgehensmodell mit Meetings, Artefakten, Rollen, Werten und Grundüberzeugungen, das beim Entwickeln von Produkten im Rahmen agiler Softwareentwicklung hilfreich ist.

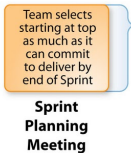
# Wikipedia

- OOPSLA 1995, erster Konferenzbeitrag
- Begriff durch Jeff Sutherland, Ken Schwaber geprägt
- Umsetzung von Lean Development für das Projektmanagement
- Kernprinzipien
  - Transparenz: Fortschritt und Hindernisse eines Projektes werden täglich und für sichtbar festgehalten
  - Überprüfung: Regelmäßige Lieferung
  - Anpassung: Anforderungen und Implementierungen entwickeln sich. Neubewertung und Anpassung nach jeder Lieferung

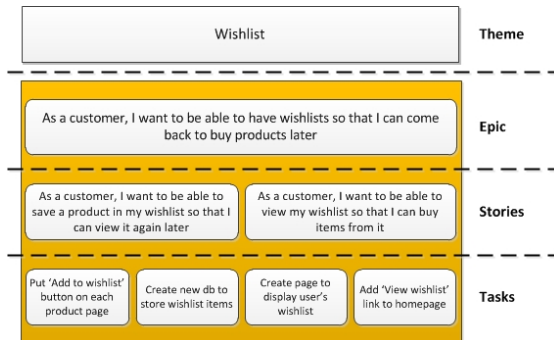
# Der Scrum Prozess

## The Agile: Scrum Framework at a glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



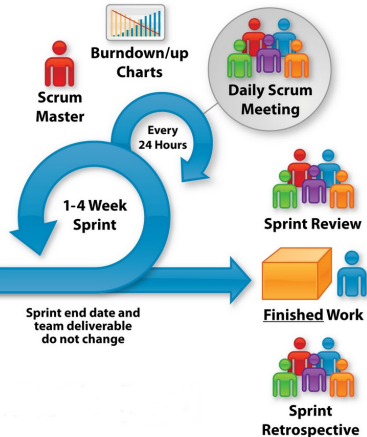
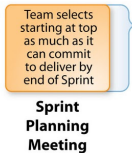
# Die Planungseinheiten



# Der Product Owner

## The Agile: Scrum Framework at a glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



# Der Product Owner

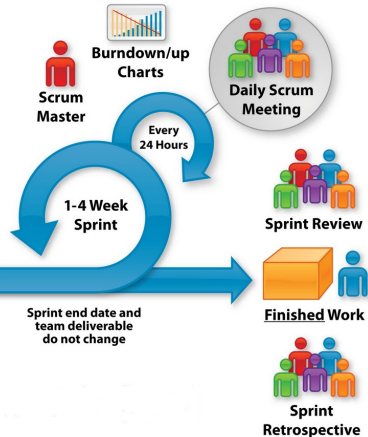
- Verwaltet das Produktbacklog
- Bringt Stories in einen bearbeitbaren Status (Erfüllung der „Definition of Ready“)
- Definiert Produkt-Features
- Priorisiert Features abhängig vom Marktwert
- Akzeptiert oder weist Arbeitsergebnisse zurück
- Bestimmt Auslieferungsdatum und Inhalt
- Ist verantwortlich für das finanzielle Ergebnis des Projekts (ROI)
- Passt Features und Prioritäten nach Bedarf für jeden Sprint an



# Das Team

## The Agile: Scrum Framework at a glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



# Das Team

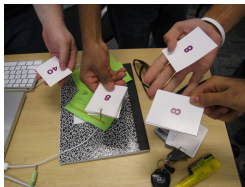
- Typischerweise 5-9 Personen
- Funktionsübergreifend / Interdisziplinär  
(QS, Programmierer, UI-Designer, System-Engineer, etc.)
- keine Titel (aber manchmal nicht vermeidbar)
- Mitglieder sollten Vollzeitmitglieder sein - wenige Ausnahmen  
(z.B. sehr selten benötigte Spezialisten)
- Mitgliedschaft kann sich nur zwischen Sprints verändern
- Teams organisieren sich selbst
- Team erfüllt die „Definition of Done“

# Scrum Planning Poker



Aufgaben werden durch das Team in einem abstraktem Storypoint-Komplexitätsmaß unabhängig von der Qualifizierung der Teammitglieder geschätzt.

# Scrum Planning Poker

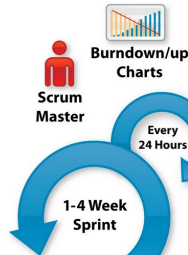
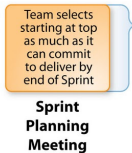


- Gemeinsame Einschätzung durch Pokern und Austausch von Argumenten
- Summe der Komplexitäten = Umfang des Sprints
- Betrachtung der Leistung des letzten Sprints und Commitment des Teams für bevorstehenden Sprint
- PO kann mit frühzeitigem Schätzen und der Betrachtung der Storypoints z.B. 1-3 Sprints vorausplanen

# Das Daily Standup

## The Agile: Scrum Framework at a glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



Sprint end date and team deliverable do not change

# Daily Standup



- Ziel: Die Zusammenarbeit koordinieren und den Arbeitsfluss optimieren (Reviews, Peering, Hindernisse beseitigen, ...)
- Unbedingt vermeiden: Technische Diskussionen, reines Statusreporting, Unpünktlichkeit bzw. Nichtanwesenheit
- Was gehört ins Daily?
  - Gestern habe ich ...
  - Heute will ich ...
  - Mich behindert ... ich komme nicht weiter an ... weil ...
  - Ich könnte Unterstützung bei ... gebrauchen

# Scrum Board I

The screenshot shows a Jira Scrum Board for the project 'Domäne I'. The board is organized into four columns: 'Aufgaben' (Tasks), 'Wird Ausgeführt' (In Progress), 'Abnahme PO' (Review), and 'Fertig' (Done). The board is currently displaying tasks from the 'Aufgaben' column.

**Domäne I**

SPRINT: 10.02. - 02.03. | SCHNELL-FILTER: Nur meine Vorgänge | Zuletzt aktualisiert | Sticky Items | User Story | Bug | Service & Support

**Aufgaben**

- DOMI-314 3 Unteraufgaben - Bestandsdaten Anpassung Berechnung Shop-Bestand
- DOMI-332 6 Unteraufgaben - Bestandsdaten Adaption Genehmigungsstatus
  - DOMI-451 Testfälle mit PM erstellen
  - DOMI-452 UnitTest
- DOMI-368 5 Unteraufgaben - Analyse Auswirkungen Auflösung auf Datenmodell in hybris

**Wird Ausgeführt**

**Abnahme PO**

**Fertig**

- DOMI-449 UnitTest
- DOMI-450 Wiederverwendung ProductStatusService
- DOMI-453 Integrationstest anpassen
- DOMI-456 Analyse bestehender Import
- DOMI-459 Doku erstellen

Dieses Board wurde aktualisiert: Aktualisieren - Ignorieren

# Scrum Board II

15.04 | Der Artikelstamm wird von der 100 an die 1000 und Hybrid mit der eConv-ID übertragen und dort gespeichert. | #2

30.04

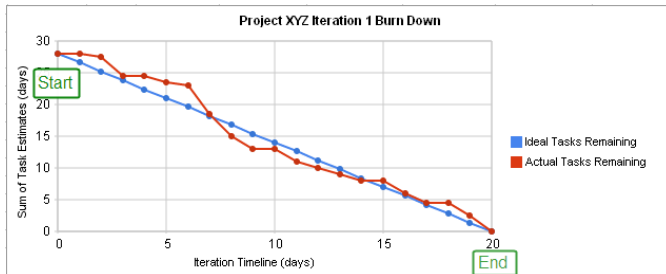
**Doing**

**Done**

101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000



# Das Burndown Chart



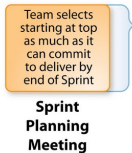
# Der Scrum Master

- Repräsentiert das Management gegenüber dem Projekt
- Verantwortlich für die Einhaltung von Scrum-Werten und -Techniken
- Beseitigt Hindernisse
- Stellt sicher, dass das Team vollständig funktional und produktiv ist
- Unterstützt die enge Zusammenarbeit zwischen allen Rollen und Funktionen
- Schützt das Team vor äußeren Störungen

# Der Scrum Zyklus

## The Agile: Scrum Framework at a glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



# Das Sprint Review Meeting

- Das Team präsentiert, was es während eines Sprints erreicht hat
- Der PO gibt einen Ausblick auf den nächsten Sprint
- Typischerweise in Form einer Demo der neuen Features oder der zugrunde liegenden Architektur
- Zwei Stunden zur Vorbereitung
- Keine Folien
- Das ganze Team nimmt teil
- Jeder ist eingeladen

# Die Sprint Retrospektive

- Nach jedem Sprint
- bei 3 Wochen Zyklus, ca. 2 Stunden
- Das ganze Team nimmt teil: Scrummaster, Produktowner, Team
- Regelmäßiger Rückblick auf Erfolge und Verbesserungsmöglichkeiten
- Nutzung methodischer Ansätze
  - Retormat
  - Tastycupcakes
- Identifizieren und Auswählen von Sprint-Team-Zielen
- Nachhalten Sprint-Team-Zielen

# Was bringt Scrum dem Fachbereich?

- Verlässlichkeit - alle 3 Wochen ein Release
- keine Change Request Diskussionen mehr
- Bessere Ergebnisse durch Orientierung am Kunden und am Benutzer
- Strukturierung in Epics und Stories vereinfacht die Priorisierung
- Transparenz und kurzfristigere Steuerung
- Priorisierung des Backlogs nach Return On Investment und somit Businesswert
- Kostenkontrolle: „Danke. Ich habe ausreichend Funktionalitäten erhalten.“
- Productowner als Anlaufstelle für Fachlichkeit, die Technik und den Zeitplan
- Produktentscheidungen zum Zeitpunkt der Relevanz
- keine Produktentscheidungen ohne Informationen

# Was bringt Scrum dem Entwicklungsteam?

- Das Team organisiert sich und seine Arbeit selbst
- Enge fachliche Zusammenarbeit mit dem Product Owner
- Konzentration und Fokussierung
- Besser funktionierende Software durch iterative Vorgehensweise
- Code wird wartbarer, Bewußtsein dass alles im Fluß ist
- Bessere Softwarearchitektur - passende Lösungen
- Anforderungen entwickeln sich sinnvoll und können mitgestaltet werden
- Kontinuierliche Verbesserung: keine Ownership, Wissenstransfer, Retros, ...
- Einfachheit - die Kunst, die Menge nicht getaner Arbeit zu maximieren
- Definition of Ready: Klare Kriterien für die Implementierungsfähigkeit einer Story
- Definition of Done: Klare Kriterien für den Abschluß einer Aufgabe

# Überblick

- 1 Einführung
- 2 Agiles Arbeiten
- 3 Kanban
- 4 Scrum
- 5 Abschluß**
  - Letztes Slide
  - Agiles Werkzeug
  - Referenzen



# Fragen ?

## Werkzeug für das agile Arbeiten:

- Ein Moderationskoffer
- Ein Wiki System  
(Teamorga, Dokumentation, Prozesse, ...)
  - [XWIKI](#)
  - [Atlassian Confluence](#)
- Ein Board  
(Arbeitsorganisation, Transparenz, Visualisierung, ...)
  - Ein physisches Board
  - [Trello](#)
- Ein skalierendes Chatwerkzeug  
(Themenbezogene Kommunikation statt Directchat)
  - [Rocketchat](#)
  - [Mattermost](#)
- Ein Team das Verantwortung übernehmen will
- Methodische Unterstützung bei der Etablierung
- Geduld und Konsequenz
- Die Akzeptanz der Stakeholder, des Unternehmens

## Literatur / Referenzen

- [Scrum – auf dem Bierdeckel erklärt](#)
- [The Toyota Way: Fourteen Management Principles from the World's Greatest Manufacturer](#)
- [Who is the Professional Scrum Master](#)
- [Scrumguide / Sutherland, Schwaber - praktisch „die Erfinder“ von Scrum](#)
- DevOps for Developers / Michael Hüttermann (Apress Verlag)
- The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win / Kim, Behr, Spafford (IT Revolution Press)
- Peopleware: Productive Projects and Team / Tom DeMarco
- [Der Sourcecode dieser Präsentation / Download](#)