



# Clean Code – Von der Lehre in den Alltag

Daniel Georges



- Head of Solutions Engineering bei Westernacher Solutions AG
- 20 Jahre Erfahrung u.a. als
  - Entwickler
  - Softwarearchitekt
  - Projektmanager / Product Owner
  - CTO
- Fokus auf Beratung
  - Agile Coach
  - Softwarecraftsman



- Unternehmensberatung mit Schwerpunkt Strategie-, Prozess- und IT-Implementierung
- Gegründet 1969
- Internationales Unternehmen mit Hauptsitz in Deutschland
- 500 Westernacher Berater
- 100% in Besitz der Berater



HEIDELBERG  
STUTTGART  
BERLIN  
DÜSSELDORF



SOPOT



BRÜSSEL



WIEN



BOSTON



LONDON




SHANGHAI



MUMBAI

# Agenda

 This image cannot currently be displayed.

Clean Code


Softwarecraftsmanship

Agile

Stakeholder

Einsatz in der Praxis

# Agenda

 This image cannot currently be displayed.

Clean Code

Softwarecraftsmanship

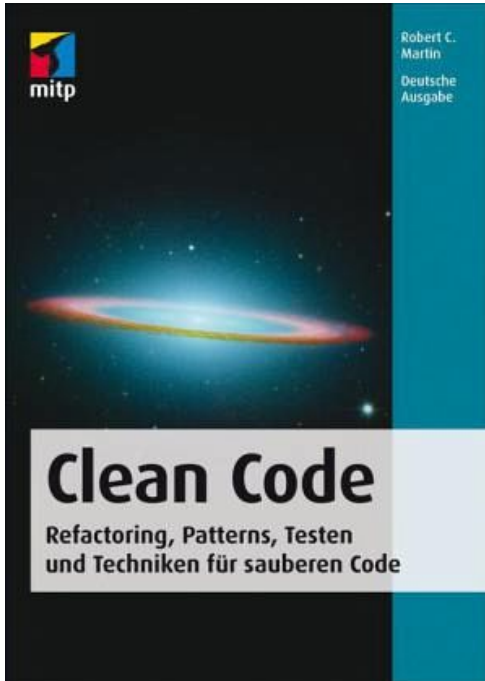
Agile

Stakeholder

Einsatz in der Praxis

# Clean Code

## Verschiedene Prinzipien und Praktiken



- Guter Code
  - ist leicht verständlich (für andere)
  - ist leicht erweiterbar (von anderen)
  - ist auf eine Sache fokussiert
  - ist durch Tests abgesichert
  - hält sich an KISS
  - besitzt möglichst wenige Dependencies
  - TDD, Patterns, DRY, YAGNI, SOLID ...

# Clean Code

## Herausforderungen in der Softwareentwicklung



- Unklare oder wechselnde Anforderungen
- Unterschiedliche Erfahrungsstände, Qualitäten oder Vorgehensweisen beim Entwickeln von Software
- Mangelnde Disziplin beim Entwickeln, u.a. forciert durch Zeit- und Kostendruck
- Hoher Aufwand nachträglicher Änderungen durch:
  - Technische Schulden
  - Fehlende Lesbarkeit / Dokumentation
  - Wechselnde Teams / Wissensträger
  - Fehlende Standards / Tooling
  - Fehlende Testabdeckung

# Clean Code

## Wie hilft uns Clean Code diese Herausforderung zu meistern?

- Unklare oder wechselnde Anforderungen
- Unterschiedliche Erfahrungsstände, Qualitäten oder Vorgehensweisen beim Entwickeln von Software
- Mangelnde Disziplin beim Entwickeln, u.a. forciert durch Zeit- und Kostendruck
- Hoher Aufwand nachträglicher Änderungen durch:
  - Technische Schulden
  - Fehlende Lesbarkeit / Dokumentation
  - Wechselnde Teams / Wissensträger
  - Fehlende Standards / Tooling
  - Fehlende Testabdeckung
- Clean Code kann einfacher erweitert / geändert werden. Durch Testabdeckung gewährleistet. KISS / MVP
- Softwaretechniken und Standards helfen. Pair Programming, Peer Code Review, Tools
- Softwaretechniken und Standards helfen. Pair Programming, Peer Code Review, Tools
- Softwaretechniken und Standards helfen
  - Refactoring / Campingplatzprinzip
  - Einfach zu lesender Code mit Namenskonventionen
  - Pairprogramming / Selbsterklärender Code
  - Standards als Konvention / Continuous Integration
  - TDD / Unit- / Integrationsstests als Bestandteil

# Agenda

 This image cannot currently be displayed.

Clean Code

**Softwarecraftsmanship**

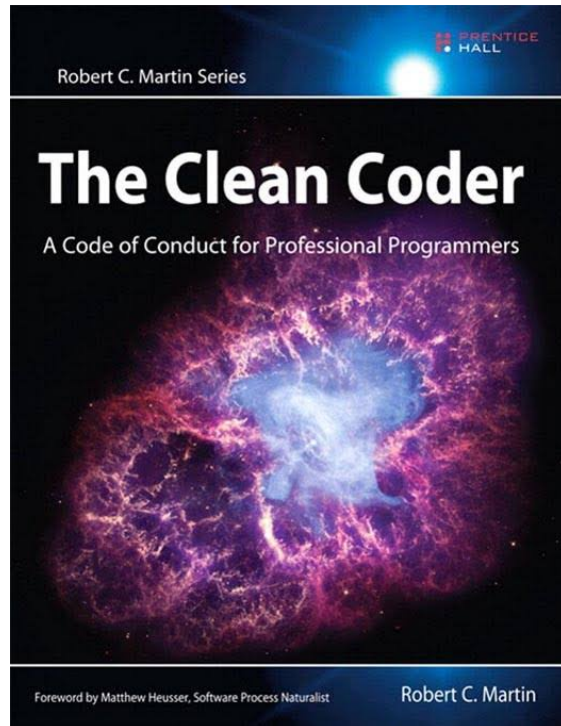
Agile

Stakeholder

Einsatz in der Praxis

# Softwarecraftsmanship

## Professionelle Einstellung zur Softwareentwicklung

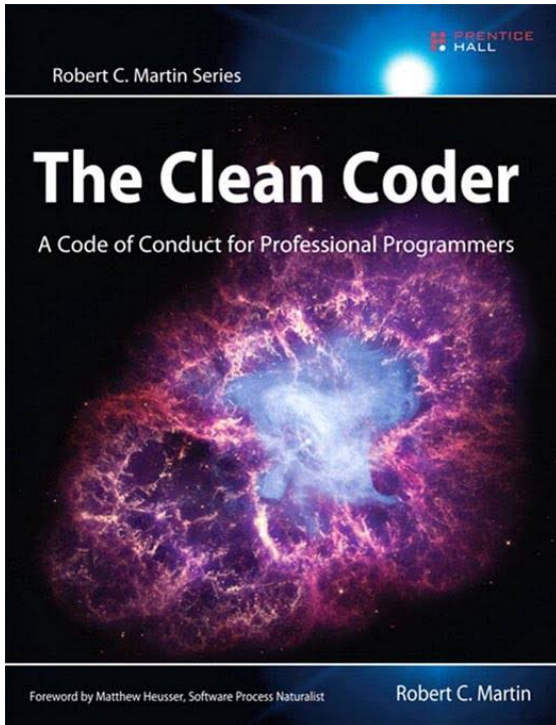


- Als engagierte **Software-Handwerker** heben wir die **Messlatte für professionelle Softwareentwicklung** an, indem wir **üben** und anderen dabei **helfen**, das Handwerk zu erlernen.
  - Nicht nur funktionierende Software,  
**sondern auch gut gefertigte Software**
  - Nicht nur auf Veränderung zu reagieren,  
**sondern stets Mehrwert zu schaffen**
  - Nicht nur Individuen und Interaktionen,  
**sondern auch eine Gemeinschaft aus Experten**
  - Nicht nur Zusammenarbeit mit dem Kunden,  
**sondern auch produktive Partnerschaften**

Quelle: <http://manifesto.softwarecraftsmanship.org/#/de>

# Softwarecraftsmanship

## Professionelle Einstellung zur Softwareentwicklung



- Softwarecraftsmanship heißt
  - Selbstverständnis für, was gute Software ist und wie dies erstellt werden kann
  - Ehrliche und offene Kommunikation
  - Lebenslanges Lernen
  - Wissensaustausch mit Anderen

# Softwarecraftsmanship

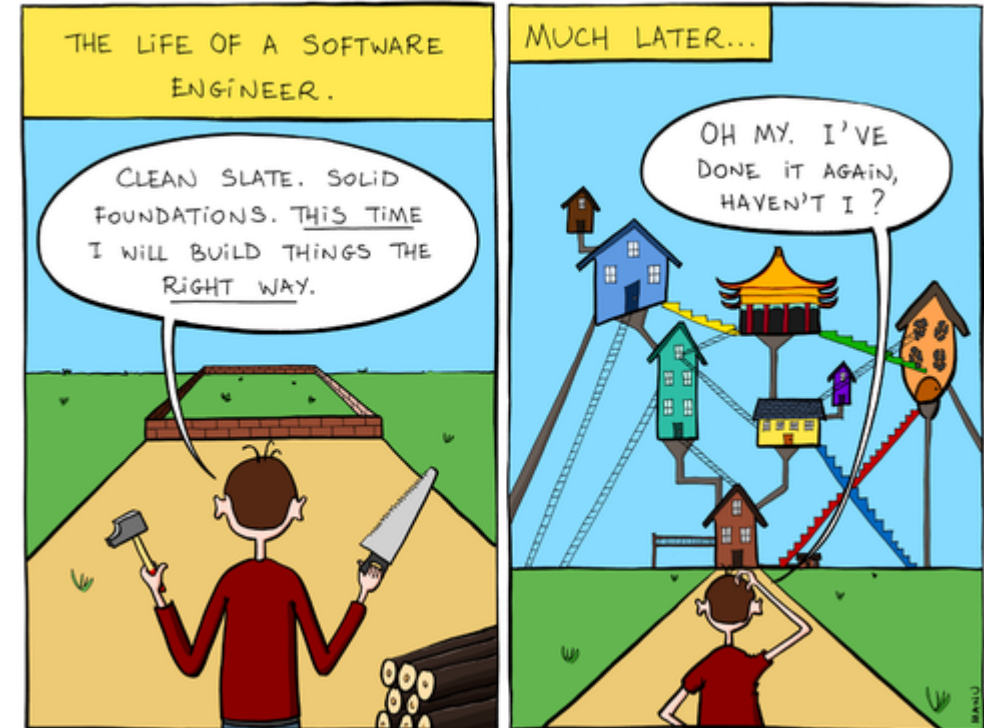
## Professionelle Einstellung zur Softwareentwicklung



Quelle: <http://www.nedelykov-moreira.de/en/denkmalpflege>




Quelle: <http://www.haus.de/bauen/bauen-mit-holz/dachstuhl-sanieren-so-klappt-es-wie-am-schnuerchen.htm>



Quelle: [https://twitter.com/objective\\_neo/status/523085425817419776](https://twitter.com/objective_neo/status/523085425817419776)

# Agenda

 This image cannot currently be displayed.

Clean Code

Softwarecraftsmanship

**Agile**

Stakeholder

Einsatz in der Praxis

# Agile Softwareentwicklung

## Werte und Prinzipien



- Our **highest priority is to satisfy the customer** through early and **continuous delivery of valuable software**.
- Welcome **changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
- **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- **Business people and developers** must work together daily throughout the project.
- **Build projects around motivated individuals**. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is **face-to-face** conversation.

# Agile Softwareentwicklung

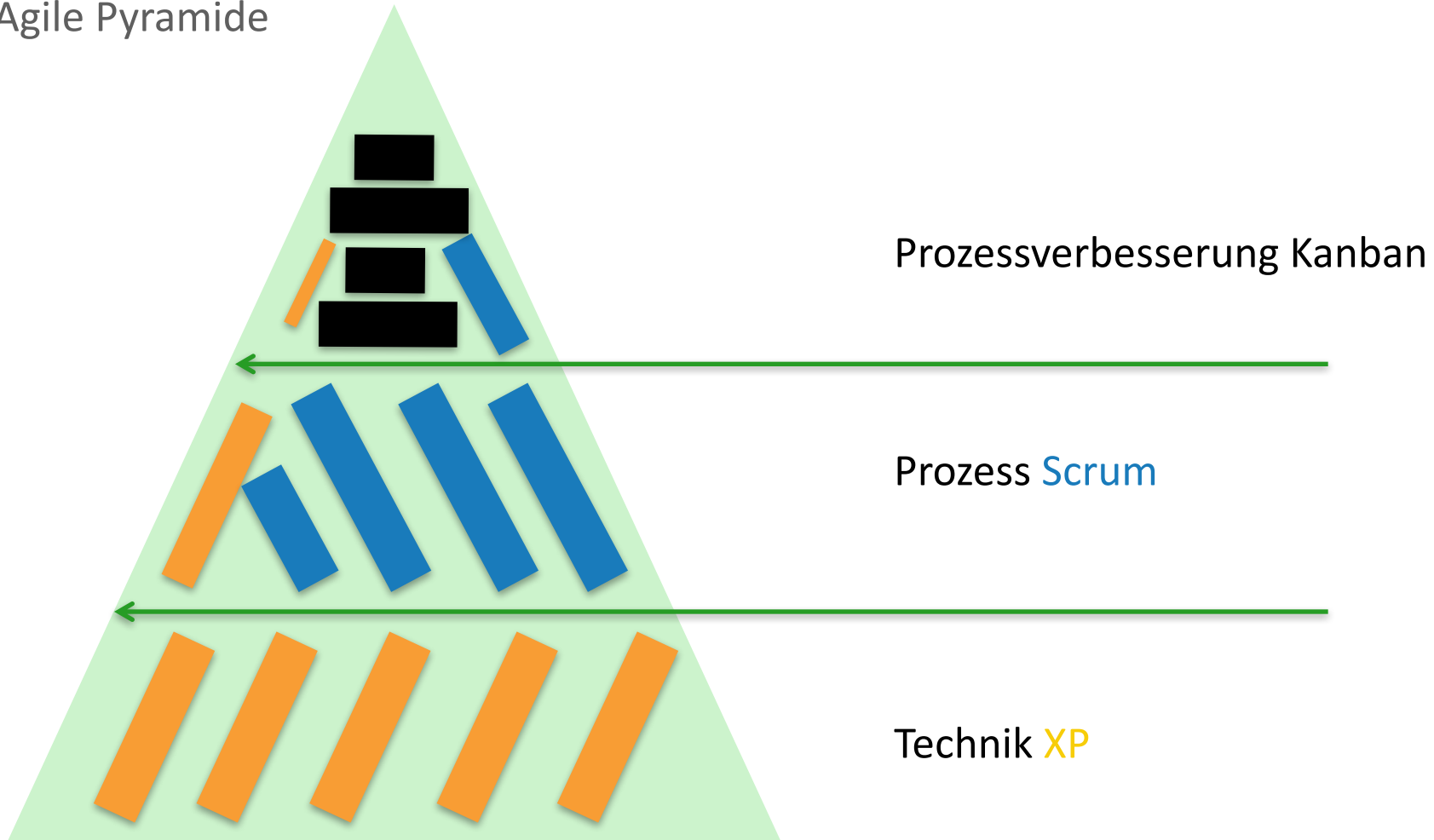
## Werte und Prinzipien



- **Working software** is the primary measure of progress.
- Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely.
- **Continuous attention to technical excellence** and **good design** enhances agility.
- **Simplicity**--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from **self-organizing teams**.
- **At regular intervals, the team reflects** on how to become more effective, then tunes and adjusts its behavior accordingly.

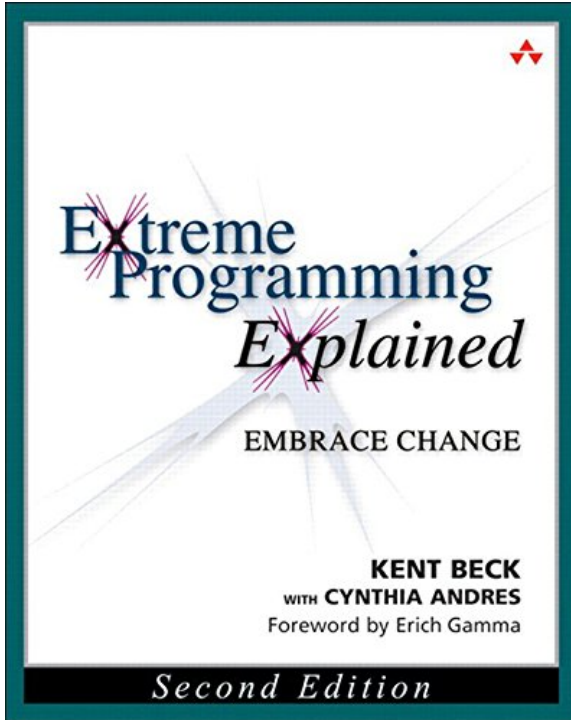
# Agile Softwareentwicklung

## Agile Pyramide



# Agile Softwareentwicklung

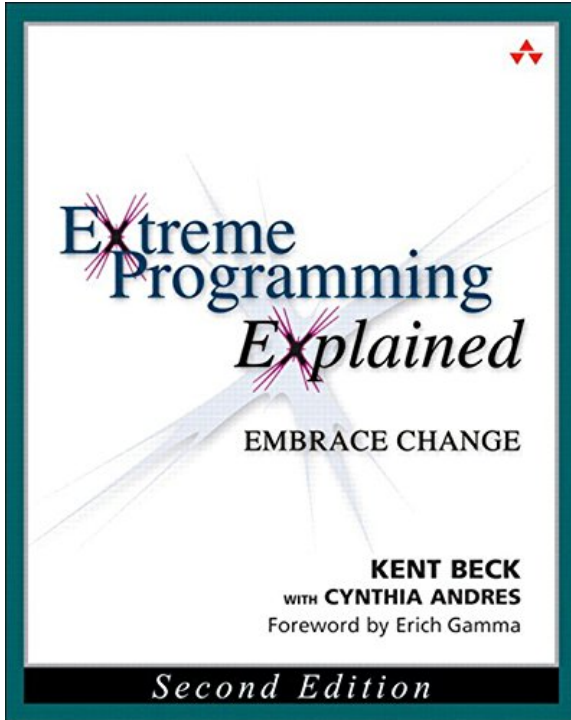
## eXtreme Programming - Werte



- Simplicity (KISS/YAGNI)
- Communication (Face-to-Face)
- Feedback (Software & Process)
- Respect (Respect the expertise)
- Courage (Honesty)

# Agile Softwareentwicklung

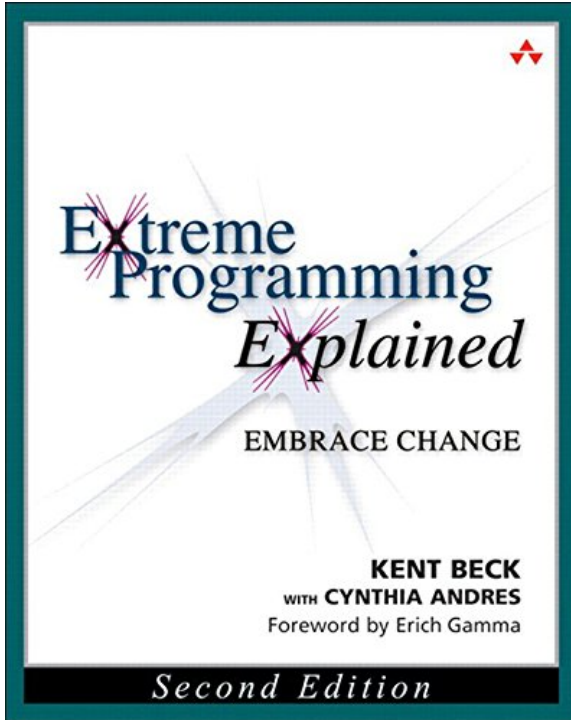
## eXtreme Programming – Rules Planning



- User stories
- Release planning
- Frequent small releases
- Project is divided into iterations
- Iteration starts each iteration

# Agile Softwareentwicklung

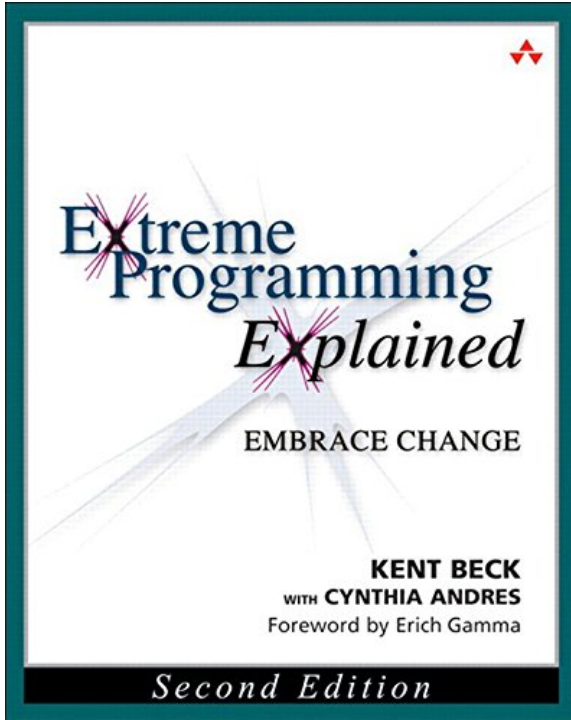
## eXtreme Programming – Rules managing



- Open work space
- Sustainable pace
- Stand up meeting each day
- Project velocity is measured
- Move people around
- Fix XP when it breaks

# Agile Softwareentwicklung

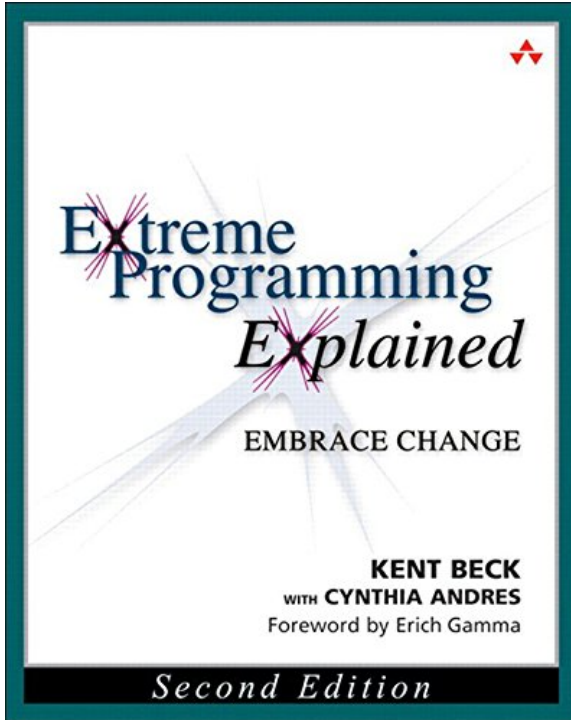
## eXtreme Programming – Rules designing



- Simplicity
- System metaphor
- CRC cards for design sessions
- Spike solutions to reduce risk
- Early functionality is prohibited
- Refactoring

# Agile Softwareentwicklung

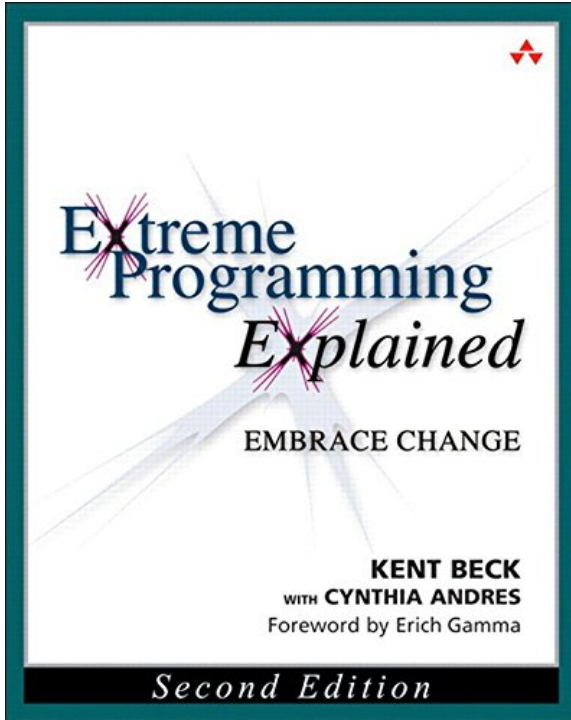
## eXtreme Programming – Rules Coding



- On site customer
- Coding standards
- TDD
- Pair programming
- Only one pair integrates code
- Integrate often
- Dedicated integration computer
- Collective ownership

# Agile Softwareentwicklung

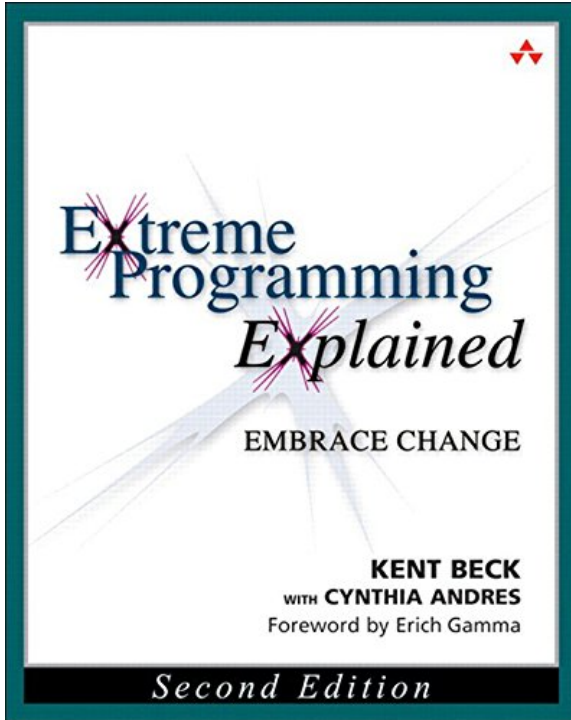
## eXtreme Programming – Rules Testing



- All code must have unit tests
- All unit tests must be passed before
- For each bug a test is created
- Acceptance tests

# Agile Softwareentwicklung

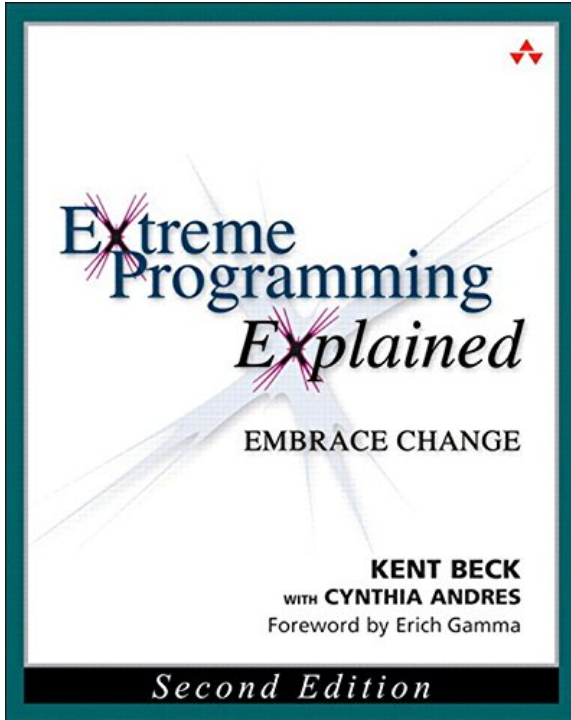
## eXtreme Programming – Fundamental principles



- Rapid Feedback
- Assume Simplicity
- Incremental Change
- Embracing Change
- Quality Work


# Agile Softwareentwicklung

## eXtreme Programming – Further principles



- Teach Learning
- Small Initial Investment
- Play to Win
- Concrete Experiments
- Open, honest Communication
- Work with people's instincts - not against them
- Accepted Responsibility
- Local Adaptation
- Travel Light
- Honest Measurement

# Agenda

 This image cannot currently be displayed.

Clean Code

Softwarecraftsmanship

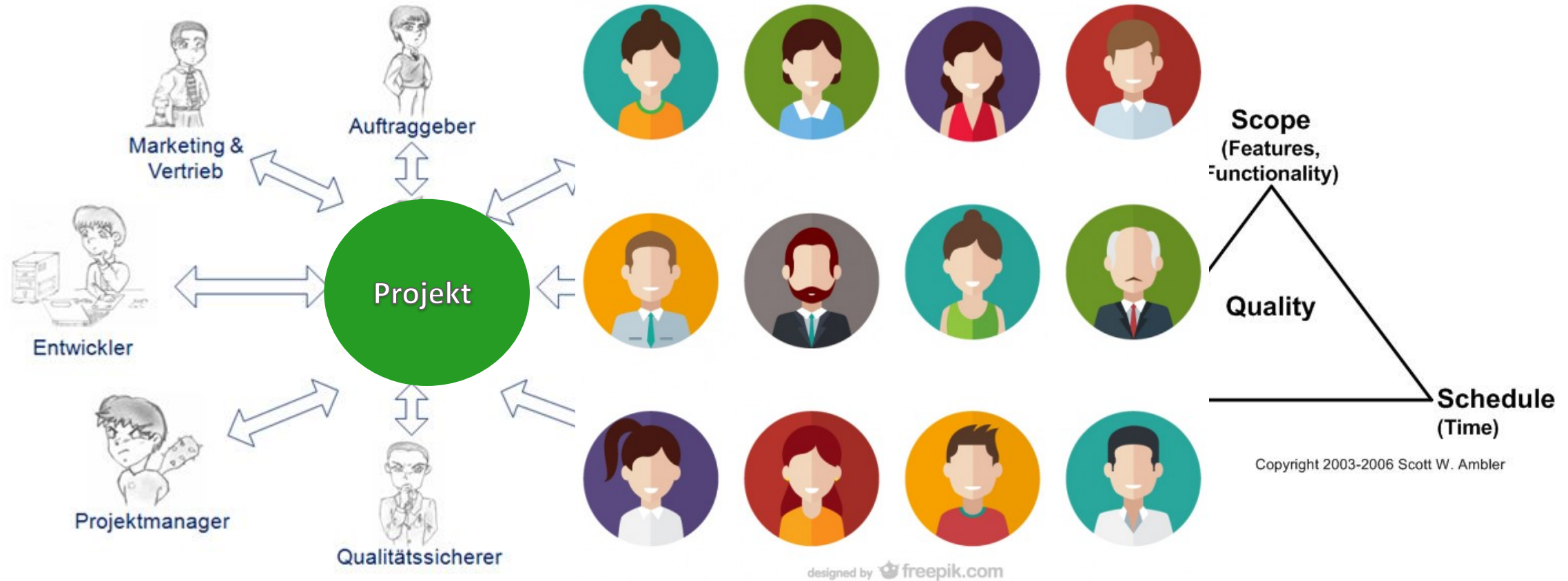
Agile

**Stakeholder**

Einsatz in der Praxis


# Stakeholder

## Stakeholder und Interessen



Quelle: <http://blog.sophist.de/2015/10/08/stakeholder-analyse-dokumentation-fundamentale-stakeholder-klassen-wie-sie-stakeholder-identifizieren/>

# Agenda

 This image cannot currently be displayed.

Clean Code

Softwarecraftsmanship

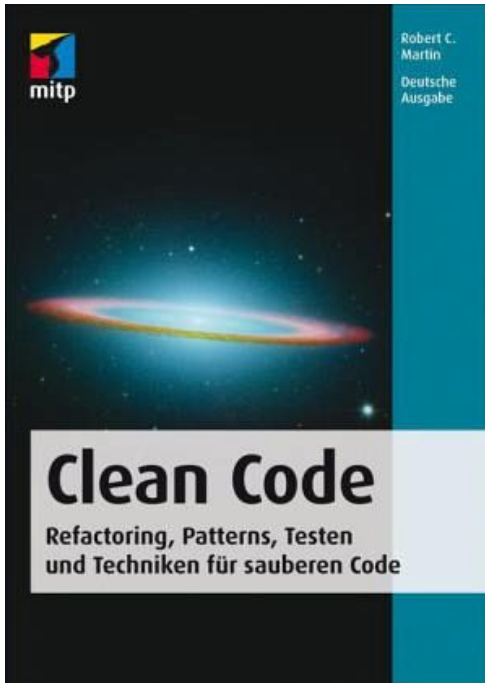
Agile

Stakeholder

**Einsatz in der Praxis**

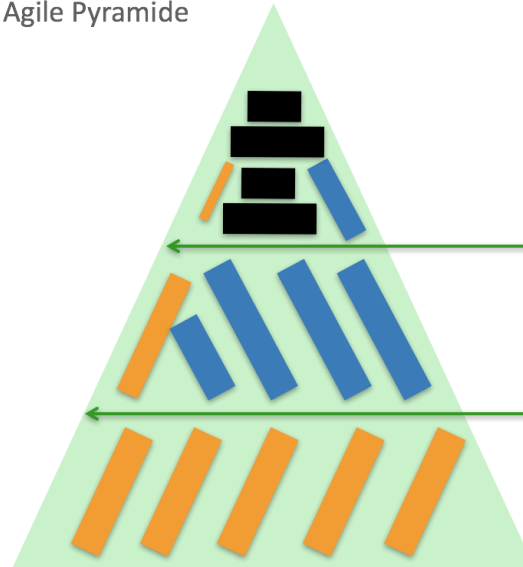
# Einsatz in der Praxis

Wie passt alles zusammen?



Professionalität

Agile Pyramide



Prozessverbesserung Kanban

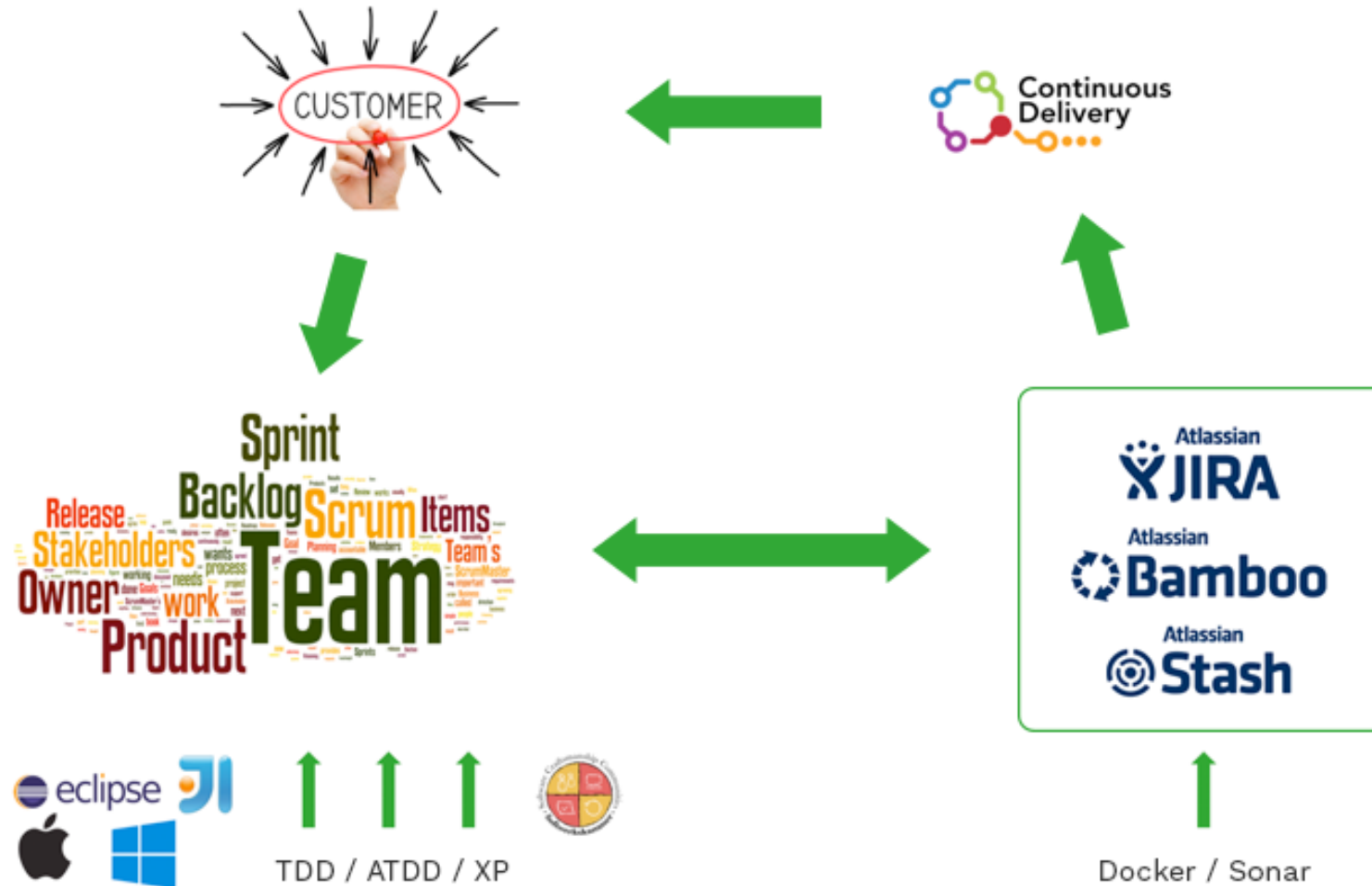
Prozess Scrum

Technik **XP**



# Einsatz in der Praxis

## Westernacher Agile Software Development Process



# Einsatz in der Praxis

## Empfehlungen



- Üben, üben, üben
- Austausch mit anderen Experten
- Anderen helfen besser zu werden
- Konzentration auf das Wesentliche KISS/MVP – Werte liefern
- Schnelle Auslieferungen – Lernen über Feedback
- Keine unrealistischen Termine zusagen – Softwarecraftsman sagen auch nein
- Die Qualität liefern, die erwartet wird

As a **Software Craftsman**

**You love what you do  
and you do it with passion**



# Clean Code – Von der Lehre in den Alltag

Daniel Georges

# Contact information

**Daniel Georges**

Head of Solution Engineering

Königstrasse 26  
70173 Stuttgart/ Germany



+49 151 64960299



daniel.georges@westernacher.com



www.westernacher-solutions.com

Alle Angaben basieren auf dem derzeitigen Kenntnisstand. Änderungen vorbehalten.

Dieses Dokument ist ausschließlich für den Adressaten bestimmt.  
Es bleibt bis zu einer ausdrücklichen Übertragung von Nutzungsrechten  
Eigentum von Westernacher Products & Services AG.

Jede Bearbeitung, Verwertung, Vervielfältigung und/oder gewerbsmäßige  
Verbreitung des Werkes ist nur mit Einverständnis von Westernacher  
Products & Services AG zulässig.

  
**WESTERNACHER**  
SOLUTIONS  
21.06.17