









Agile Manifesto



We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

INDIVIDUALS AND INTERACTIONS—over processes and tools WORKING SOFTWARE—over comprehensive documentation CUSTOMER COLLABORATION—over contract negotiation RESPONDING TO CHANGE—over following a plan

That is, while there is value in the items on the right, we value the items on the left more



Agile Manifesto

have

PROCES

VE DOC

TRACT

Values

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland Dave Thomas

Signatories

We follow these principles:

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.

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.

17 signatories 2001 agilemanifesto.org

Principles

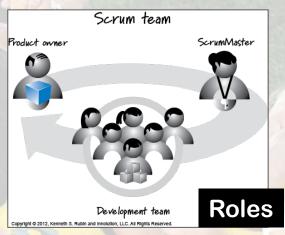
Scrum



Scrum

"Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems."





Commitment, Focus, Openness Respect Courage

Values

Basis for most iterative processes / methods currently in use.

Opinion

Transparency Inspection Adaption

Theory

Most popular method (66% uptake)

Good range of materials and certification

Priority on business value









Certification is subject of debate
No technical practices specified
Small team focused
Numerous sources

Jeff Sutherland / Ken Schwaber 1996 scrumguides.org

1a

ScrumBut (ScrumAnd)



ScrumBut (ScrumAnd)

"We're doing Scrum but..."
"We use Scrum and..."



ScrumAnd

ScrumAnd framework

"We're doing Scrum but..."
our sprints are 12 weeks long"

2006

(3)

ScrumBut





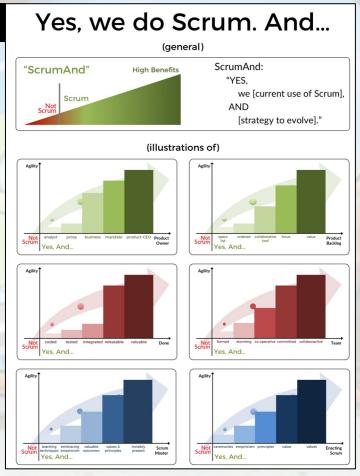
"We use Scrum

and we are collaborating

and brainstorming within the

Scrum Team to increase

value every Sprint."



Eric Gunnerson Ken Schwaber

https://docs.microsoft.com/en-us/archive/blogs/ericgu/scrumbut

2008 kenschwaber.wordpress.com/2012/04/05/scrum-but-replaced-by-scrum-and/

2

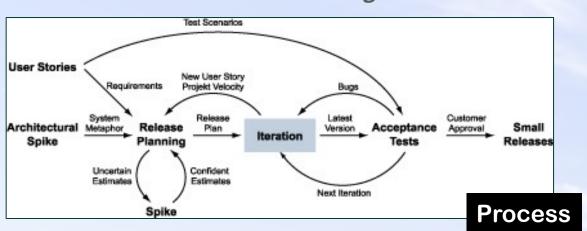
Extreme Programming (XP)

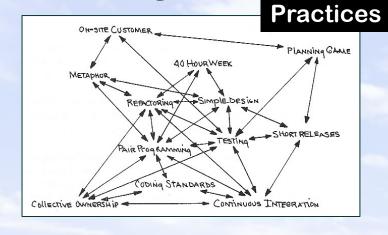


Extreme Programming (XP)

"Extreme Programming is the first popular methodology to view software development

as an exercise in coding rather than an exercise in management."





Principles

Feedback **Assuming Simplicity Embrace Change**





Practices have become software engineering good practice

Communication Feedback Simplicity Courage Respect

Values

Opinion

Focus on code **Delivers** quality Reduced risk

Small Team focus Requires craftsman Require discipline

Kent Beck 1996 extremeprogramming.org

Crystal



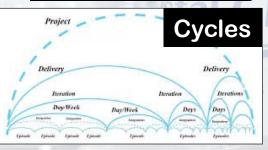
Crystal

"Crystal is a family of human-powered, adaptive, ultralight, "stretch-to-fit" software development methodologies... designed to scale by project size and criticality"

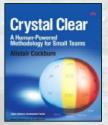
3	Clear	Yellow	Orange	Red	Magenta	Blue
Life	L6	L20	L40	L100	L200	L500
Essential	E 6	E20	E40	E100	E200	E500
Discretionary	D6	D20	D40	D100	D200	D500
Comfort	C6	C20	C40	C100	C200	C500
	1-6	20	40	100	200	500

Exploratory 360 degree
Early victory
Walking skeleton
Incremental
rearchitecture
Information radiators

Methodology: Size vs Criticality







Frequent delivery

Reflective improvement
Osmotic communication
Personal safety
Focus

Easy access to expert users
Technical environment
(automated tests, configuration
management, frequent
integration

Properties Strategies

Accounts for life critical products
Cross functional teams
Human powered

Hard to change size
Requires co-located team
Lightweight, requires
discipline

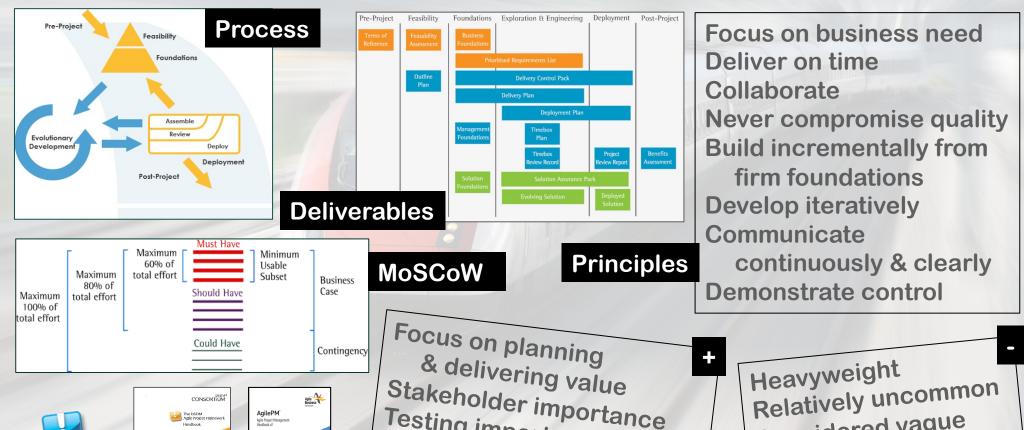
Alistair Cockburn 1992 alistair.cockburn.us

Dynamic Systems Development Method (DSDM) (AgilePM)



Dynamic Systems Development Method (DSDM) (AgilePM)

"Incorporates project management disciplines... to ensure... project benefits are clear,... proposed solution is feasible and ... solid foundations in place..."









Stakeholder importance Testing importance Evolved to AgilePM in 2016

Considered vague in places

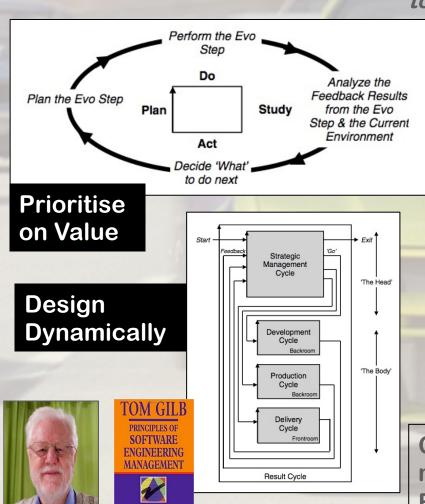
DSDM Consortium (now Agile Business Consortium) 1994 agilebusiness.org

Evolutionary Project Management (Evo)



Evolutionary Project Management

"Focus on delivering measurable multiple value requirements to stakeholders."



Principles

Decompose by performance results and stakeholders; Do high-risk steps early, learn how 'unknowns' really

Focus on improving your most valuable performance objectives first;

Base your early evolution on existing systems and stakeholders;

Design to cost dynamically;

Design to performance dynamically;

Invest in an open-ended architecture early on;

Motivate your team by rewarding results;

Prioritize changes by value, not place in queue;

Learn fast, change fast, adapt to reality fast.

Considered original agile method

Focus on value delivered to stakeholders

Limited Resources (mostly available via training) Newer methods have built on principles

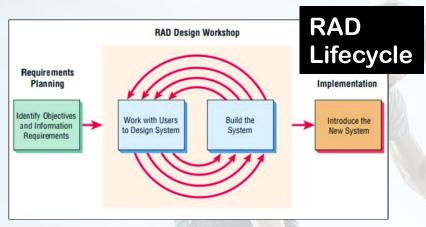
Tom Gilb 1960 gilb.com

RAD (Rapid Application Development) / Adaptive Software Development (ASD)



RAD (Rapid Application Development) / Adaptive Software Development (ASD)

"RAD is designed to give mush faster development and higher quality results" "ASD does not provide prescriptive rules... but a framework of concepts, practices..."

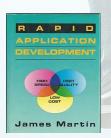


High quality High Speed RAD Characteristics

Low Cost / Maintenance Meet needs of users









adaptive cycle planning
uses mission statement
project constraints
basic requirements
time-boxed release plan

Requirements gathering
JAD
mini-specs

collaboration

Release
software increment
adjustments for subsequent cycles
components implemented/tested
focus groups for feedback
formal technical reviews
postmortems

Mission focused Feature based Iterative Time-boxed Change tolerant Risk driven

ASD Characteristics

Introduced iterative phases
Focus on human relationships
with users
RAD very successful in Australia



Uncommon / dead (but antecedents to newer approaches)
Concept of "good enough" work
RAD considered "hack & test"

James Martin → Jim Highsmith 1991 → 1992



Lean / Lean Manufacturing / Lean Enterprise / Toyota Production System



Lean / Lean Manufacturing / Lean **Enterprise / Toyota Production System**

"Lean uses less of everything compared with mass production... As it inevitably spreads... will change almost everything in almost every industry..."

Lean **Principles**

Toyota Production System



Improvement Kata

Understand the Direction

8 Wastes (Muda)

Underutilizing

people's talents.

skills, & knowledge,

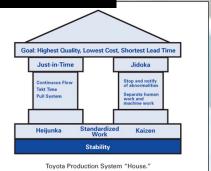
Grasp the Current Condition



Establish the **Next Target** Condition











Excess products





Wasted time waiting



movements by people (e.g., walking).









Unnecessary movements of products & materials.



More work or higher quality than is required



more than needed or

Many learnings for manufacturing, IT & business in general

Manufacturing focused Can be difficult to adapt to knowledge work Constant gathering of metrics

Eli Whitney → Taichi Ono → James Womack 1850 → 1936 → 1990

lean.org

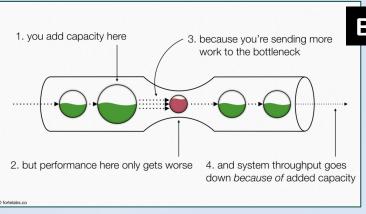
7a

Theory of Constraints



Theory of Constraints

"A chain is no stronger than its weakest link."



Bottlenecks

Thinking Process

Gain agreement on the problem Gain agreement on the direction for a solution Gain agreement that the solution solves the problem Agree to overcome any potential negative ramifications Agree to overcome any obstacles to implementation

Theory of Constraints **Throughput**

The Goal



Basis of many methods & models Many applications Well regarded



Steps

Some parts still theoretical

7b Lean Software Development



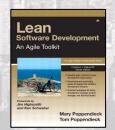
7b Lean Software Development

"Lean uses less of everything compared with mass production...
As it inevitably spreads... will change almost everything in almost every industry..."

7 Principles / 22 Tools

	Lean Principle	Description		
D	Deliver Fast	Deliver value to the customer quickly, rapid delivery; high quality; low cost Queuing theory to Limit Work in Process (WIP) and context switching Manage workflow is easier than managing schedules, using repeatable workflow		
E	Eliminate Waste	Waste is anything that does not add value to the customer. The three biggest wastes in software development are: 1. Building the wrong thing: building features that aren't needed 2. Failure to learn: policies that interfere with our ability to learn 3. Thrashing: anything that interferes with smooth flow of value		
L	Learn Constantly	Predictable performance is driven by feedback: rapidly respond to change Maintain options; keep code change tolerant, minimise irreversible decisions Defer commitment, schedule irreversible decisions to Last responsible moment		
	Build Quality In (Integrate Quality)	Final Verification should not find defects! Prevent with executable requirements Mistake proof your process with test first development to establish correctness Break dependencies: architecture should support addition of any feature at any time		
V	Optimize the Whole (Value the Whole)	Focus on the entire value stream from customer request to deployed software Deliver a complete product, a complete team delivering not just the software Think long term rather than local optimization		
E	Engage Everyone	Autonomy: Empowered self-organising feature teams with effective leadership Mastery: Provide challenge and environment which enables people to grow Purpose: Tie the work to value and a common vision		
R	Keep Getting Better (Relentless Improvement)	Failure is a learning opportunity: investigate and correct them as they occur Standards exist to be challenged and improved Use the scientific method Plan-Do-Check-Act process		





Applied lean manufacturing techniques to software development

Following books deal with leadership & mindset Techniques high level

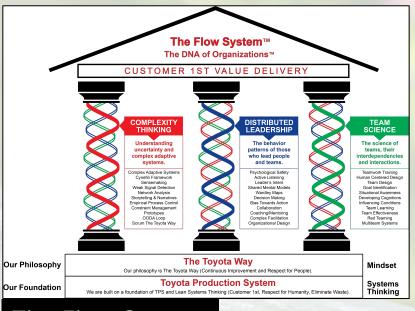
7c

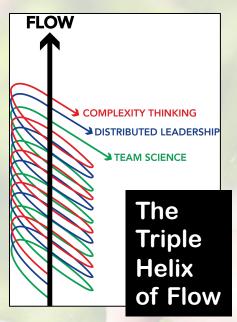
The Flow System



The Flow System

"A re-imagined system for organizations to understand complexity, embrace teamwork, and leverage autonomous team-based leadership structures."





Customer 1st

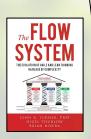
Highest Quality
Lowest Cost
Shortest Lead Time
FLOW of value
The Triple Helix of Flow
Complexity Thinking
Distributed Leadership
Team Science

The Flow System









thinking for VUCA world
Builds upon lean thinking and TPS

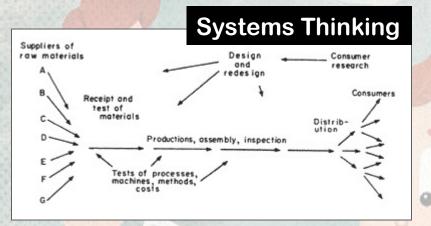
Flow Guide reasonably basic
Based on existing tools
Comprehensive
certification /
assessment

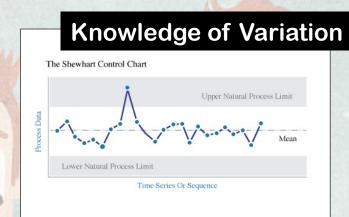
Deming System of Profound Knowledge

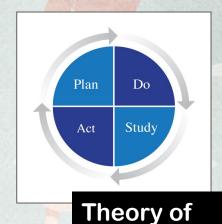


Deming System of Profound Knowledge

"An effective theory of management that provides a framework of thought and action for any leader wishing to transform and create a thriving organization..."





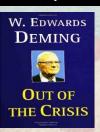


Knowledge



Principles





Deming's 14 Points					
1	Create constancy of purpose .	8	Drive out fear.		
2	Adopt the new philosophy and take on leadership .	9	Break down barriers. Work as a team.		
3	Eliminate inspection. Build in quality.	10	Eliminate slogans. Fix the system.		
4	Minimize total cost of by improving quality of supplies.	11	Eliminate quotas. Substitute Leadership		
5	Constantly improve quality and productivity to decrease costs.	12	Remove barriers to pride of workmanship.		
6	Institute training on the job.	13	Institute a vigorous program of education and self-improvement.		
7	Supervision should be to help people.	14	The transformation is everybody's job.		

Requires organizational change
Focus on internal process

Builds quality in Simplified process

Requires organiza

W. Edwards Deming

1939 deming.org

for management

14 points

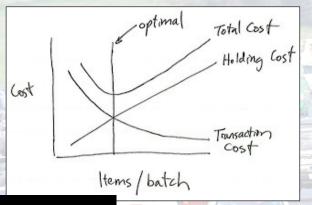
9 Product Development Flow

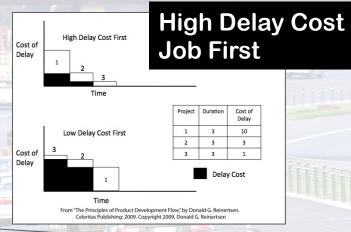


Product Development Flow

"...the dominant paradigm for managing product development is wrong.

Not just a little wrong, but wrong to its very core."



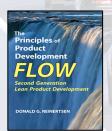


Economic
Queueing
Variability
Batch Size
WIP Constraint
Flow Control
Fast Feedback
Decentralization

Optimum
Batch Size

12 Cardinal Sins





Failure to correctly quantify economics
Blindness to queues
Worship of efficiency
Hostility to variability
Worship of conformance
Institutionalization of large batch sizes
Underutilization of cadence
Managing timelines instead of queues
Absence of WIP constraints
Inflexibility
Noneconomic flow control
Centralized control

175 Principles

Proves why Agile / Lean works
Scientific evidence

Can be hard to understand
Does not provide
specific techniques

Donald G. Reinertsen

2009

lpd2.com

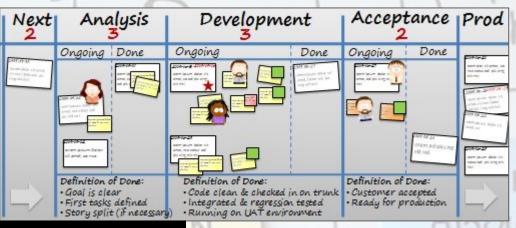
10

Kanban

Doing ToDo Done GET SOME LEARN USE STICKY KANBAN ABOUT NOTES! KANBAN TRY GETA KANBAN WHITE -TOOL BOARD

Kanban

"...evolutionary change model that utilizes a kanban (small k) pull system, visualization, and other tools to catalyze the introduction of Lean ideas into technology..."

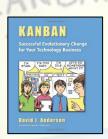


Start with what you do now
Agree to pursue incremental,
evolutionary change
Respect the current process, roles,
responsibilities and titles
Leadership at all levels
4 Principles

Kanban Board

5 Core Properties





Visualize workflow
Limit work-in-progress
Measure & manage flow
Make process policies
explicit
Use models to recognize
improvement opportunities

Simple
Bottlenecks visible
Removes iterations for
reactive work
Focus on WIP

Requires discipline
Often misunderstood
Fracturing community

David J. Anderson 2010 kanban.university

10a

Personal Kanban



Personal Kanban

"...provides a light, actionable, achievable framework for understanding our work and its context."



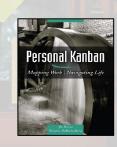
Get your stuff ready
Establish your value stream
Establish your backlog
Establish your work in progress limit
Begin to pull
Reflect
5 Steps

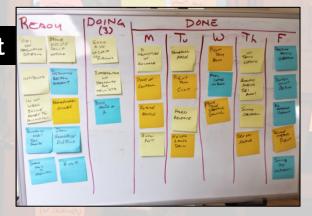
2 Rules of Personal Kanban

Throughput









Simple personal focus
Personal context, clarity
& closure
Focus on WIP

Little guidance on scaling
Focus on individual

Lean Startup



Lean Startup

"It's ultimately an answer to the question 'How can we learn more quickly what works, and discard what doesn't?"



5 Principles

Entrepreneurs are everywhere

Entrepreneurship is management

Validated Learning

Build - Measure - Learn

Innovation Accounting

Agile assumed at core

intrapreneurs

delivery

Guidance for startups or

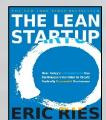
Learn through continuous

NYT #2 Bestseller

Popularised

Minimum Viable Product (MVP) **Continuous Deployment Split Testing Actionable Metrics** Pivot **Innovation Accounting**

Lean / Business Model Canvas



Hype Focus on features not products

Eric Ries 2008 theleanstartup.com

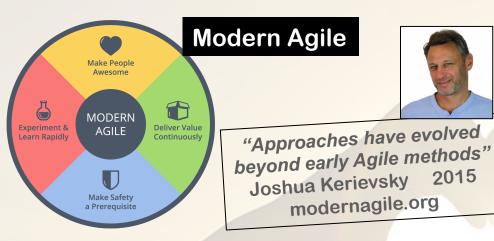


Modern Agile / Heart of Agile / Agnostic Agile / Value Lifecycle



Modern Agile / Heart of Agile / Agnostic Agile / Value Lifecycle

"Modern interpretations of agility"





Heart of Agile

"Amplifies agile culture by simplifying it" Alistair Cockburn heartofagile.com/

Agnostic Agile

"Agility that is adaptive within your given system according to context and is not by necessity a prescription of any particular framework or method" Sam Zawadi agnosticagile.org/

I seek to uphold the following principles, to the best of my ability and judgment TO RESPECT

TO PUT MY CUSTOMER TO RECOGNISE THAT FIRST, MAKING THEM FRAMEWORKS AND

INDEPENDENT THEIR PRACTITIONERS TO DO MY BEST, COMPLEMENTING THEOR

TO ACKNOWLEDGE **UNKNOWNS AND**

PRINCIPLES

THERE IS MORE TO AGILE THAN AGILE

WITH PRACTICAL SEEK HELP **EXPERIENCE**

TO GIVE TO THE COMMUNITY AS IT HAS **GIVEN TO ME**

TO TAILOR AGILITY TO TO NEVER MISLEAD AND

TO EMBRACE DIVERSITY, WITHOUT

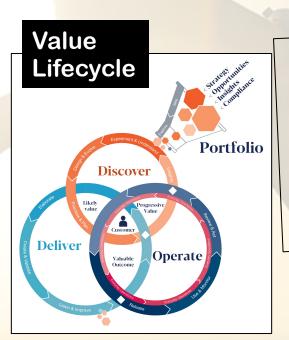
TO UNDERSTAND HINDERING CONSTRAINTS AND WORK TO REMOVE THEM

TO REMEMBER THAT **AGILITY IS NOT THE END**

TO SHARE, LEARN AND

TO ACKNOWLEDGE THAT **DOGMATISM IS NON-AGILE**

🚰 agnostic Agile



"The Agile lifecycle is made up of continuously spinning phases -Discover, Deliver and Operate" Craig Smith / 2020 SoftEd softed.com





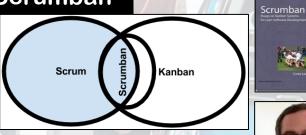
Hybrid Agile (Scrumban / Xanpan / ScrumXP / Water Scrum Fall / BOSSA nova)



Hybrid Agile (Scrumban / Xanpan / ScrumXP / Water Scrum Fall / BOSSA nova)

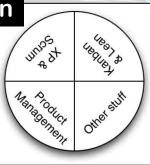
"Using one or more aspects of different methods..."

Scrumban



Cross between Scrum & Kanban
Corey Ladas 2008
agilealliance.org/scrumban/

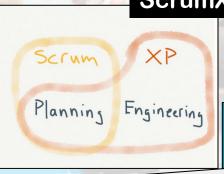
Xanpan



Cross between Kanban & XP – plus Scrum & Lean Alan Kelly 2014 allankelly.net/books/#xanpan

ScrumXP

SAFe



Combining team and technical agility
Scaled Agile 2016
scaledagileframework.com/scrumxp

Water Scrum Fall



Manage the boundaries to increase agility & see the benefits

Dave West 2011 forrester.com Combining Beyond
Budgeting, Open
Space, Sociocracy
& Agile

Jutta Eckstein / John Buck 2016 agilebossanova.com



BOSSA nova

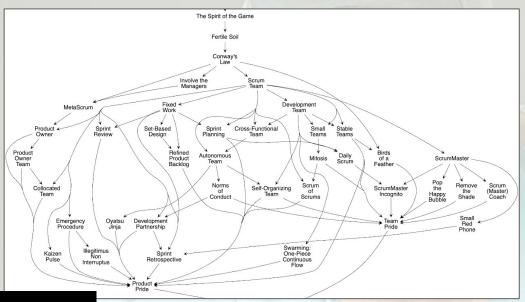


ScrumPLOP (Pattern Languages of Programs)



ScrumPLOP (Pattern Languages of Programs)

"ScrumPLoP mission is to build a body of pattern literature around... communities, describing... insights, so we can easily share them."



Pattern Name		Patlet	Author	Date	
	Sprint Retrospective		Neil Harrison and Mike Beedle	2013-07-02	Sprint
Sprint Backlog		The team needs a list to guide production work; Therefore: Make a Sprint Backlog that lists all work in a Sprint	Mike Beedle and Ademar Aguiar	11/11/2010	Value Stream
	Sprint Backlog Item	The above patterns talk about the ordering of the work plan, being free on an SBI level and chuncked at a PBI level. Within PBIs developers can create or delete tasks at will	Dina Friis & Jens Østergaard	2013-03-28	Value Stream
	Dependencies First	Dependencies suck. Therefore: Handle them in the first half of the Sprint and abort the Sprint if you can't.	James O. Coplien	2012-06-16	Value Stream
	Swarming: One-Piece Contin	Development Team has the final say over the ordering of Sprint backlog items. One-piece continuous flow should be split out of this.	Jeff Sutherland	2013-03-13	Value Stream
	Developer-Ordered Work Pla	People want direction, but task lists decrease self-organization; Therefore: Work from an unordered sprint backlog	James O. Coplien	2010-05-01	Value Stream
	First Things First	Pattern on working the top of the backlog (破)	Jeff Sutherland	2013-03-13	Value Stream
	Work the Top of the Back	Pattern on working the top of the backlog (破)	Jeff Sutherland	2013-03-13	Value Stream
	Burndown Chart	The obvious	Alan O'Callaghan	2015-01-05	Value Stream
	Track DONE	It's easy to misinterpret the burn-down chart overly optimistically with respect to business value; Therefore: Track business value progress separately and in additional to incremental task completion — REMOVED and factored into Sprint Burdown	James O. Coplien	2010-05-01	Value Stream
<u>Definition of Done</u>		You cannot achieve quality if everyone has a different definition of quality. Therefore: the team defines criteria of all work to be completed on usual work items.	Ville Reijonen	9/18/2015	Value Stream
	Automated Code Checks	Definition of DONE does not mean "Ready to Ship" Solution : Work on automating checks of the code	Jens Østergaard	5/1/2010	Misc
	Definition of Done as Work	Team members must adhere to "Done" but also have to get their work done. Therefore: Integrate "Done" with the process, so the process tends to create results that adhere to the definition.	Ville Reijonen	3/2/2012	Team

Pattern Spreadsheet

Pattern Map







Sharing rationales & experiences in a pattern form
Open for submissions

Not widely contributed to Small number of contributors
Annual review

Jeff Sutherland / Jim Coplien

2010

scrumplop.org

Agile2



Agile2

"The Agile experiment was only a partial success:
Agile now needs to pivot!"

Values

VALUES

- 1. Thoughtfulness and prescription
- 2. Outcomes and outputs
- 3. Individuals and teams
- 4. Business understanding and technical understanding
- 5. Individual empowerment and good leadership
- 6. Adaptability and planning

PRINCIPLE CATEGORIES

- 1. Planning, Transition, and Transformation
- 2. Product, Portfolio, and Stakeholders
- 3 Data
- . Frameworks and Methodologies
- 5. Technical Dimension and Technical Fluency
- 6. Individuality v. Team
- 7. Team v. Organization
- 8. Continuous Improvement
- 9. Focus
- 10. Leadership

Principle Categories

Culture Scale Technology Leadership Transformation **Product** Collaboration

Fresh look at meaning of agility
High leadership focus
Broadens agility

Lots of principles (43)
Difficult to implement due
to lack of guidance
New, low adoption

Characteristics





Adrian Lander
Cliff Berg
Huet Landry
Kurt Cagle
Lakshmi Chirravuri

Lisa Cooney MC Moore Murray Robinson Navneet Nair Parul Choudhary Philippa Fewell
Priya Mayilsamy
Raj Nagappan
Vigneshwaran Kennady
Vincent Harris

Authors

15 authors

2020

agile2.net

Extreme Manufacturing (Wikispeed) / Agile Hardware (Tesla)



Extreme Manufacturing (Wikispeed) / Agile Hardware (Tesla)

"Reduce New Production Introduction and New Product Development times by using Agile methods"

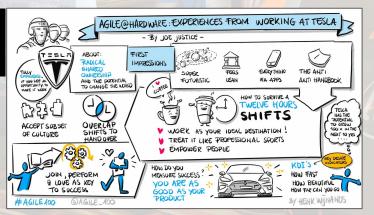
Characteristics



Built on basics of Scrum / Agile
Continuous Testing
Radical transparency & autonomy
Leadership by Example
Dynamic scaling
Focus on bottlenecks

Wikispeed

Agile@ Tesla



Applying technology practices to manufacturing Innovation – 27 changes a week at Tesla Agile outside of IT Everybody works the manufacturing line (even Elon Musk)



Not well documented – mostly videos

Hard to change factory configuration fast

Speed makes it hard for suppliers to keep up

Joe Justice 2008 wikispeed.org

Scaling Methods



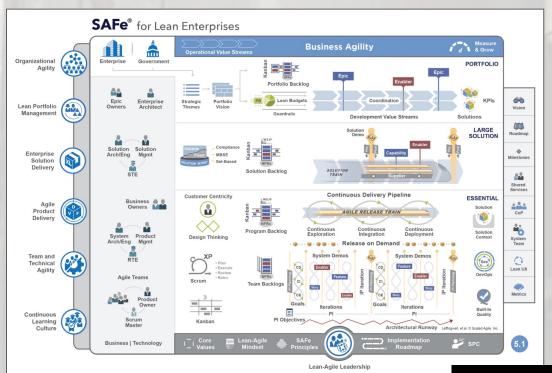
Scaled Agile Framework (SAFe)



Scaled Agile Framework (SAFe)

"...Interactive knowledge base for implementing agile practices at enterprise scale...

Highlights the individual roles, teams, activities & artefacts necessary to scale agile..."



#1 Take an economic view
#2 Apply systems thinking

1. Built-In Quality

2. Program Execution

3. Alignment

4. Transparency

#1 Take an economic view
#2 Apply systems thinking

#3 Assume variability; preserve options
#4 Build incrementally with fast, integrated learning cycles
#5 Base milestones on objective evaluation of working systems

#6 Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7 Apply cadence, synchronize with cross-domain planning

#8 Unlock the intrinsic motivation of knowledge workers

#9 Decentralize decision-making
#10 Organizie around value

Values & Principles

Most popular scaling method (37% uptake)
Big picture
Large organisation,
government & vendor buy-in









Framework

Community criticism on framework & case studies
Heavy RUP influence
Certification heavy

Dean Leffingwell

2007

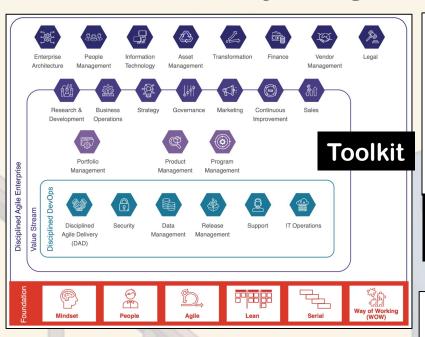
scaledagileframework.com

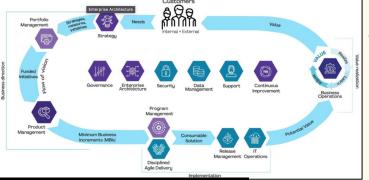
Disciplined Agile (DAD) / DA FLEX



Disciplined Agile (DAD) / DA FLEX

"...People-first, learning-oriented hybrid agile approach... has a risk-value lifecycle, is goal-driven, is scalable, and is enterprise aware."





Coverage of governance, DevOps, architecture Enterprise IT aware

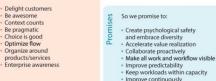
DA FLEX Lifecycle / Value Stream











We believe in these principles:

- And follow these guidelines:
- Validate our learnings
 Apply design thinking
 Attend to relationships
- through the value stream

 Create effective environments that foster joy

 Change culture by improving

Mindset

- Change culture by improving the system
- Create semi-autonomous self organizing teams
- Adopt measures to improve
- Leverage and enhance organizational assets

OProject Management Institute. All rights reserved.

Considered
heavyweight –
RUP influence
Low market
adoption
Still being
integrated by
PMI

Scott Ambler Al Shalloway

20122018

pmi.org/disciplined-agile/toolkit/value-streams

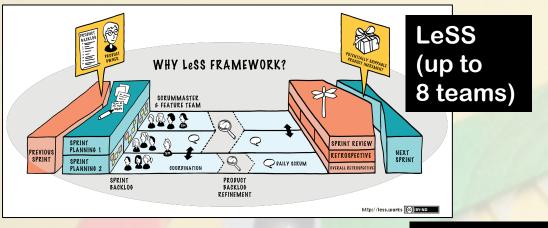
Large Scale Scrum (LeSS)



Large Scale Scrum (LeSS)

PEOPULIT OWNER

"...a label to imply regular Scrum plus the set of tips we have seen work in large multi-team, multi-site and offshore agile development."



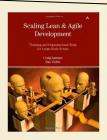
Principles

Large Scale Scrum is Scrum
Transparency
More is less
Whole product focus
Customer centric

Continuous improvement Lean thinking Systems thinking Empirical process control Queuing theory











Based on Scrum
Strong Product Owner scaling
Includes management, structure &
technical guidance
Expanded to include basic rules

SCRUM MASTER

Small adoption, but growing Can be difficult to implement as not a process

LeSS Huge (up to a

O PAILY SCRUM

few thousand people)

SPRINT REVIEW

Craig Larman / Bas Vodde

 $2008 \rightarrow 2015$

less.works



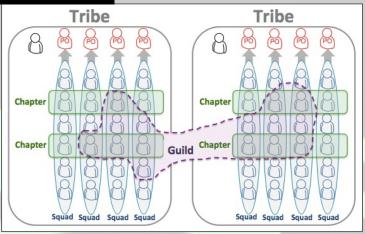


Spotify / Squadification

"...Scaling using tribes, squads, chapters and guilds"

Engineering Culture

Approach







Health Check



Principles

Loosely coupled autonomous squads
Cross pollination
Squad code accountability
Motivated individuals
Community over structure
Decoupled releases
Failure recovery
Experiment driven development

Works well for Spotify
Well regarded in community
Distributed teams

Limited documentation (PDF + videos) Not a framework, sharing experiences

Henrik Kniberg / Spotify

2012

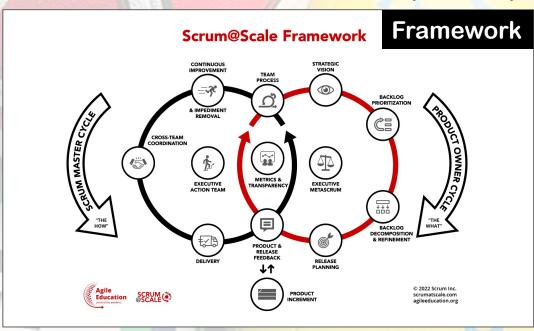
engineering.atspotify.com

Scrum@Scale



Scrum@Scale

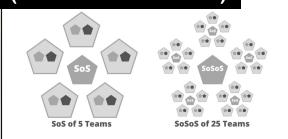
"...a lightweight organizational framework in which a network of teams operating...
...with... Scrum... can address complex adaptive problems, while... delivering... value."



Scrum of (Scrum of Scrums)







Jeff Sutherland 2014 scrumatscale.com

Small Teams (3-9 people)
Linear scalability
Minimum Viable Bureaucracy (MVB)
Scrum of Scrums
Scrum of Scrums Master /
Executive Action Team (EAT)
Chief Product Owner /
Executive MetaScrum Team (EMT)

Builds on Scrum
Defines both the Scrum Master
& Product Owner cycle
Clear information flow

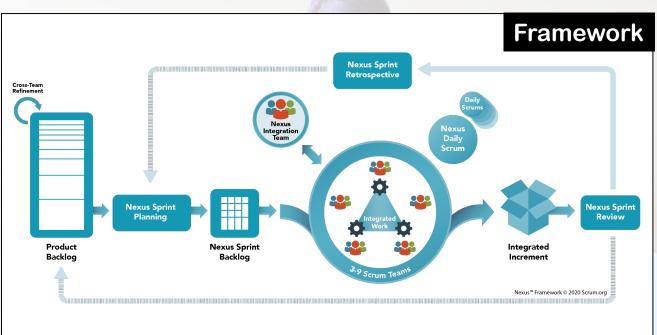
Requires organizational re-design and buy-in Low adoption but gaining interest

Nexus Framework



Nexus Framework

"...a framework consisting of roles, events, artifacts, and techniques that bind and weave together the work of approximately three to nine Scrum Teams..."



Core Features

3-9 Teams
One product backlog
Nexus Integration Team
Integrated Increment
Adds "Nexus" to the
events







Builds on Scrum
Does not consider
wider organisation
Maximum of 9 teams /
100 people

Not a lot of reported usage Limited guidance Assumes just Scrum

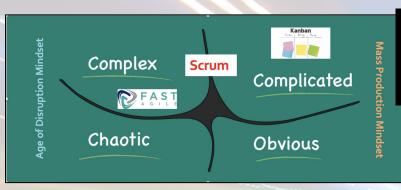
Ken Schwaber

2015

scrum.org/resources/scaling-scrum

FAST Agile (Fluid Scaling Technology)

"...combines Open Space Technology and Open Allocation to create a lightweight... method for organising people around work - that scales."



Principles

Mentor and be mentored Be a T-shaped generalising specialist

Do the right thing

Emergent design and architecture Law of mobility

2015

Complex Agile Method

Values

Autonomy
Shared Purpose
Mastery
Technical Excellence
Collaboration
Self-organisation

Pillars

Self-organising Systems
Open Space Technology
Open Allocation
Theory Y Governance
Agility
Lean Startup
Self-management



Ron Quartel



Based on dynamic reteaming
Teams form around work
One meeting – FAST meeting
Built around modern approaches

New model so low adoption currently Lacks a model / diagram for quick understanding Requires discipline

https://www.fastagile.io

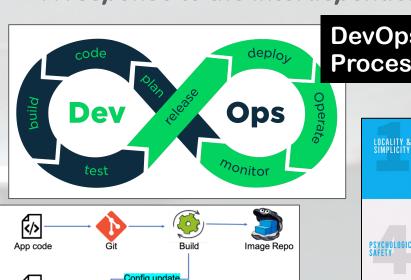


DevOps / DevSecOps / GitOps

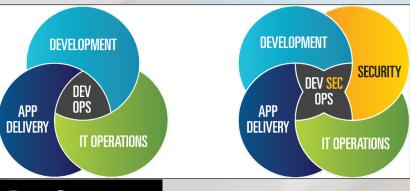


DevOps / DevSecOps / **GitOps**

"A response to the interdependence of software development and IT operations."



DevOps **Process Five** Ideals FOCUS, FLOW & JOY LOCALITY & SIMPLICITY CUSTOMER FOCUS



₹/> Cluster 1 Infra definition о-о **₹**/> GitOps tool 田 App infra definition **GitOps**

DevOps vs **DevSecOps**

First Way - performance of the entire system Second Way - create right to left feedback loops, shorten &

Third Way - creating a culture of continual experimentation & mastery

Principles





jedi.be/blog

Patrick Debois 2009

Faster downstream releases & time to market Better collaboration Large community

No definitive source Mistaken as role / department/ toolset or NoOps

MMMMMMMMMMMMMh+

Programmer Anarchy / Chaos Development

-/ `sdNMM` -/ / MMMMMMMMMMMMMMMMM MMMMMMMMMMMMMMMMM hMMMM+ sMMMmy: /MMMMm .+/-` – MMMMMMMMMMMMMMMMMM .:/os- oMMMd hNd+-: MMMMMMMMMMMMMMMMM ∇ MMMMMMMMMMMMMMMMM /MN. mMMMMMMMMm IMMMMMMMMMMMMMMMMy^

. mMMMd.

_NMMMMMMMMMMM—

.hs+. -sMMMMMMMMMMMMMM

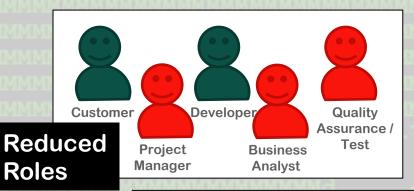
(a,b)

`my

Programmer Anarchy / Chaos Development

"Developer driven development"

"Techniques that readily accommodate in requirements during a project"



Write code that can change

Delivery focus

Enhance productivity

constantly

Fears are rampant

From Agile to Fuzzy

Microservices

Reverse Conways Law

Agile Manifesto & XP Values

Standups Trust with co location

Story narratives

Retrospectives

Estimates

Iterations Results, not blame

Mandatory pairing

Unit tests, acceptance tests

Refactoring

Patterns Small, short lived apps

Continuous integration Continuous deployment

Mantra

Lack of managers
Lightweight (beyond Agile)
Works well in right culture /
environment

Limited reference material –
short paper +
conference talks
Requires highly skilled /
disciplined team

No testing or planning

From Agile

to Fuzzy

Practices

Fred George

2010

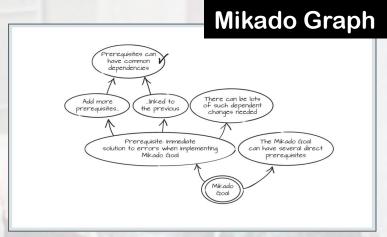
Mikado Method



Mikado Method

"...a pragmatic, straightforward, and empirical method to plan and perform non-trivial technical improvements on an existing software system."





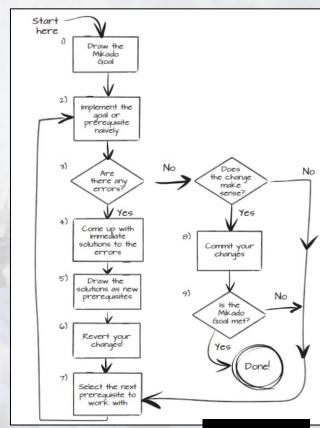
Approach to technical debt Process for effective refactoring Language tools that draw graphs







Developer focussed
Requires improvement
time



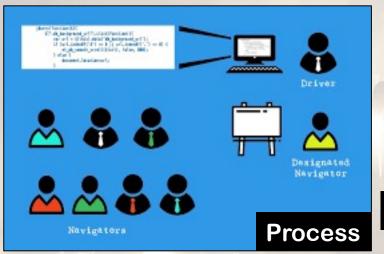
Method

Mob Programming



Mob Programming

"All the brilliant people working at the same time, in the same space, at the same computer, on the same thing"



Patterns

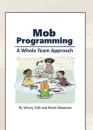


Team Rotation

Helps group organize

Better outcomes through extreme teamwork
Continuous improvement outcome
Lots of case studies, documentation, videos, books





Worked for one team initially Disciplined process Difficult to sell to traditional management

Woody Zuill 2012 mobprogramming.org

Group work area
One computer for
programming, all can see
Driver/navigator

15 minute rotations

Team communication

ownership

Individual + team +

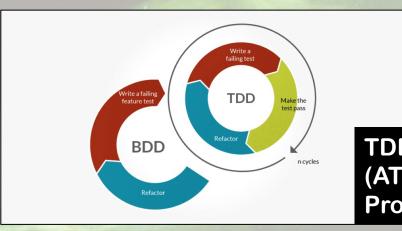
lean flow = Flow++

TDD / ATDD / BDD / SBE



TDD / ATDD / BDD / SBE

"Begin with the end in mind..."



SBE Practices

TDD / BDD (ATDD) Process Deriving scope from goals
Specifying collaboratively
Refining specifications
Automating tests based on examples
Validating frequently using tests
Evolve documentation system from
specifications with examples



Scenario / Specification

Tester
Developer

Fixture (Code)

Implementation

Scenario / Specification

Report / Living Documentation

System Under Test

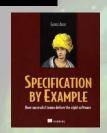
Builds quality / coverage in Provides early collaboration Tool agnostic

Structure









Requires close collaboration
Discipline and technical
expertise required
Misunderstanding of terms

Kent Beck → Dan North → Gojko Adzic

 $1994 \rightarrow 2006 \rightarrow 2011$

Context Driven Testing



Context Driven Testing

"...Advocates testing in a way that conforms to the context of the project, as opposed to testing in a way that follows some fixed notion of "best practice."

- 1. The value of any practice depends on its context.
- 2. There are good practices in context, but there are no best practices.
- 3. People, working together, are the most important part of any project's context.
- 4. Projects unfold over time in ways that are often not predictable.
- The product is a solution. If the problem isn't solved, the product doesn't work.
- Good software testing is a challenging intellectual process.
- 7. Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products.

Principles













Aligned to agile testing
Considers testing a human skill,
not a process

Not a technique, but school of thought Somewhat closed community



30

Business Agility



Business Agility

"The capacity and willingness of an organization to adapt to, create, and leverage change for their customer's benefit"



Domains of Business Agility



"An operating model for the next generation of organizations" Evan Leybourn 2017 businessagility.institute



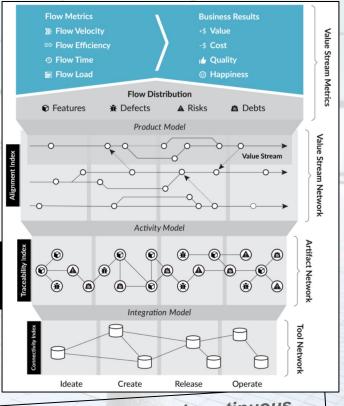
The Flow Framework



Framework for Business Agility

"Five elements that can combine to make an organisation truly agile" Andrew Craddock 2016 agilebusiness.org





"Flow, feedback, and continuous learning—we need to scale the ways of DevOps beyond IT to the business" Mik Kersten 2018 flowframework.org

Beyond Budgeting



Beyond Budgeting

"... 'Budgeting' is not used in its narrow sense of planning and control, but as a generic term for the traditional command and control management model"

Targets & rewards

Evolution

Beyond Budgeting Model: 2 x 6 Principles

Principles		Do this!	Not that!
Leadership	1. Values	Bind people to a common cause	Not a central plan
	2. Governance	Govern through shared values and sound judgement	Not detailed rules and regulations
	3. Transparency	Make information open and transparent	Don't restrict and control it
	4. Teams	Organize around a network of accountable teams	Not centralize functions
	5. Trust	Trust teams to regulate and improve their performance	Don't micro-manage them
	6. Accountability	Base accountability on holistic criteria and peer reviews	Not on hierarchical relationships

Management Processes	7. Goals	Find self-adjusting dynamic goals, relative to competitors/peers	Not short-term negotiated targets		
	8. Rewards	Base rewards on relative performance	Not fixed targets		
	9. Planning	Make planning a continuous and inclusive process	Not a top-down annual event		
	10. Coordination	Coordinate interactions dynamically	Not through annual budgets		
	11. Resources	Make resources available just-in-time	Not just-in-case		
	12. Self-Controls	Base self-controls on fast, frequent feedback	Not on budget variances		

ctions dynamically	Not through annual budgets	
vailable just-in-time	Not just-in-case	
on fast, frequent feedback	Not on budget variances	

Based on "The Leader's Dilemma	" (2011) by Jeremy Hope,	, Peter Bunce and Franz Röösli
--------------------------------	--------------------------	--------------------------------

rewards Planning & Fixed annual plans Continuous planning controls Variance controls KPI's & rolling forecasts Resource & Pre-allocated resources Resources on demand coordination Central co-ordination Dynamic coordination Central control Local control of Organisational culture Focus on managing goals/plans numbers Focus on value creation **Matrix** Processes to traditional budgeting

Traditional Budgeting

Management Model

Incremental targets

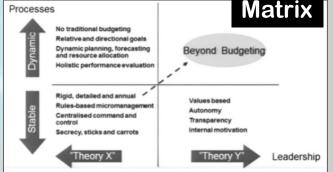
Fixed incentives

Bevond Budgeting

Management Model

Stretch goals

Relative targets &



Reduces management complexity **Implements** usable measures & financial controls Successful examples















Paid access required to join round table Requires significant senior cultural changes

Jeremy Hope / Robin Fraser / Peter Bunce → Bjarte Bogsnes

1998

bbrt.org

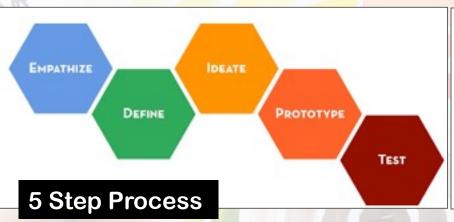
31

Design Thinking / Human Centered Design

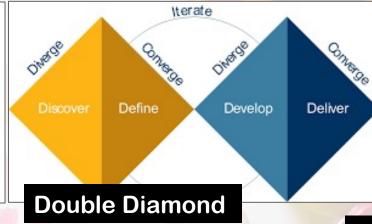


Design Thinking / **Human Centered Design**

"...a human-centered approach to innovation... to integrate the needs of people, the possibilities of technology, and the requirements for business success"

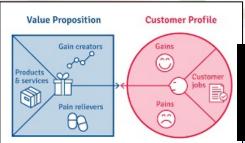


Systems **HCD~UCD** Intelligence



Human Centered Design

Customer first > Syncs with the "agile priority" to satisfy the customer Lots of materials, templates & examples



Value **Proposition** Canvas







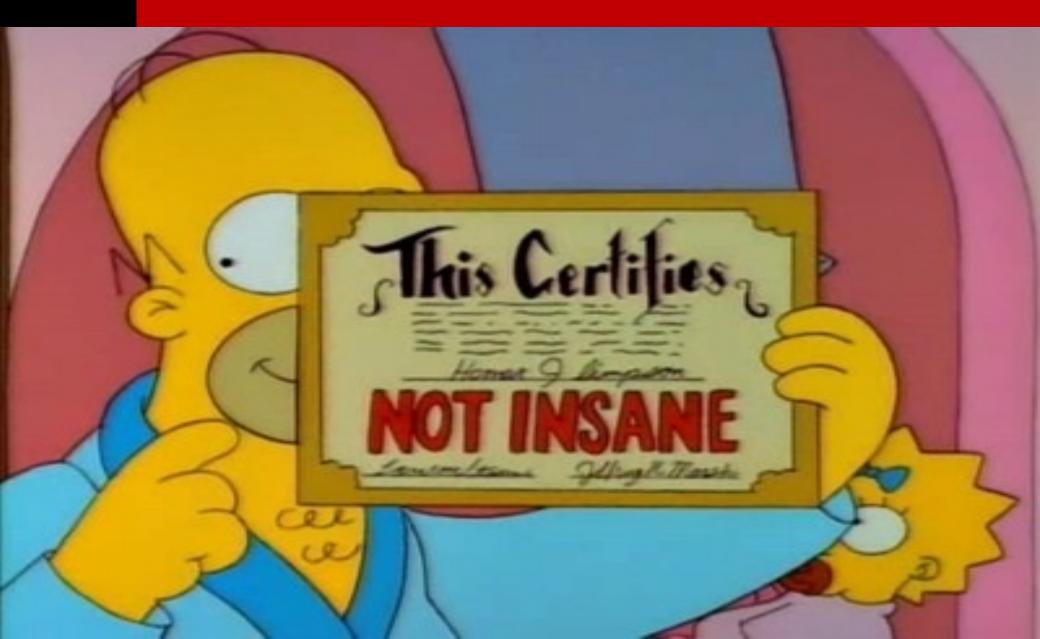




Often misunderstood terms & approaches Assumes process is linear



Role Based Certifications



Role Based Certifications



PMI Agile Certified Practitioner

"Recognizes your knowledge of agile principles and your skill with agile techniques"
Project Management Institute pmi.org

"...practical skills to incorporate Agile behaviours and techniques into PRINCE2 projects to manage projects in a way that combines responsiveness and flexibility with quality and stability."

Axelos axelos.com



PRINCE2
Agile
Practitioner

"Addresses both necessity and demand for agile and analyst communities to collaborate and transform project delivery."
International Institute of Business Analysis iiba.org



IIBA Agile Analysis ISTQB Certified Tester Foundation Level - Agile Tester



"...able to adapt existing testing experience and knowledge to Agile values and principles and support the Agile team in planning test-related activities."

ISTQB istqb.org

ICAgile Certified Professional



"...learning programs that build the capabilities necessary for business agility."

International Consortium for Agile icagile.com

Management 3.0



Management 3.0

"...Help... grow and transform organizations into becoming great places to work."



Complexity Motivation and Engagement Management and Leadership Success Delegation and and Individuals and Interactions Empowerment Failure Diversity in Collaboration The Meaning & **Happiness** Purpose Rewards & of Workers Incentives Creativity and Innovation Scaling Values & Culture Organizational Structure

Delight everybody
Improve everything
Engage People
Managing the system,
not the people
Co-create

Principles

CONSTRAINTS

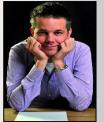
IMPROVE

EMPOWER TEAMS

Model

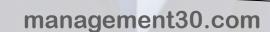
Workshop Practices

Large amount of techniques & practices Increased innovation & morale





Requires managers to have initiative Techniques in workshops, book focused on theory



34 Coaching & Facilitation



34

Coaching & Facilitation



Agile Coach Competency Framework



"...a method for technical coaches...
learning hour & ensemble working"
Emily Bache 2021
sammancoaching.org

Technical Agile Coaching

"...a framework for the competencies an Agile Coach needs..." Lyssa Adkins / Michael Spayd 2011 agilecoachcompetencyframework.com

Agile Coaching Growth Wheel



"What are the skills needed to successfully coach teams and organizations in Agile?
Scrum Alliance 2018
scrumalliance.org



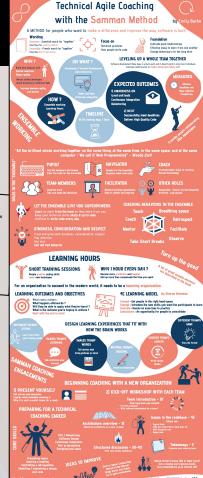
Liberating

Appendix Appendix



Samman

Method





"Simple change methods that anybody can use to improve or change the way work gets done"

Henri Lipmanowicz / Keith McCandless 2013

liberatingstructures.com/

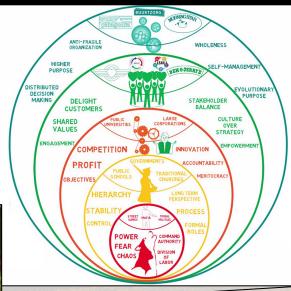


Organisation / Government Agility



Organisation / Government Agility

Reinventing Organizations



Government Agility Model





Government Agility Model

| Citizen | Centricity | Portfolio | Management | Design | Culture | Design | Culture | Culture

Day to Day Operations

Enabling Technology Agile Values & Principles

© ① S ② Designed by Craig Smith and Julian Smith
This work is licensed under the creative commons attribution-share Alike 4.0 unported licen

"In Evolutionary-Teal, we cross the chasm and learn to decrease our need to control people and events."

Frederic Laloux 2014 reinventingorganizations.com



organizations

"The 14-point standard that government services must meet."

UK Government Digital Service 2012 gds.blog.gov.uk

Service Standard

Meeting users' needs

- 1 Understand users and their need:
- 2 Solve a whole problem for users
- 3 Provide a joined up experience across all channels
- 4 Make the service simple to use
- 5 Make sure everyone can use

Providing a good service

- 6 Have a multidisciplinary te
- 7 Use agile ways of working
- Iterate and improve frequently
 Create a secure service which
 protects users' privacy
- 10 Define what success looks like and publish performance data

Using the right technology

- 11 Choose the right tools and
- 12 Make new source code open
- 13 Use and contribute to open standards common components and patterns
- 14 Operate a reliable service

www.gov.uk/service-manual/service-standard

"A framework that you can use to assess where to apply agility in your agency."

Julian Smith / Craig Smith 2014

governmentagilitymodel.com

Government
Digital Service
Standard

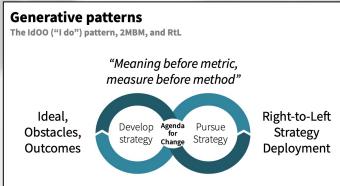
AgendaShift



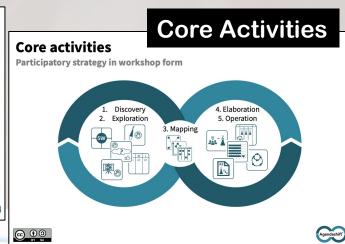
AgendaShift

"...growing wholehearted and deliberately adaptive organisations."





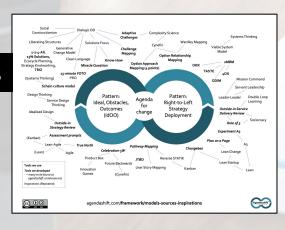
Generative Patterns



Inspirations







Open transformation framework
Workshop guidance at each core activity
Encourages engagement and focused
on outcomes
Framework agnostic

Agendashift

Generative so lacks some depth in places
Assume future states and obstacles
can be identified

Mike Burrows

2015

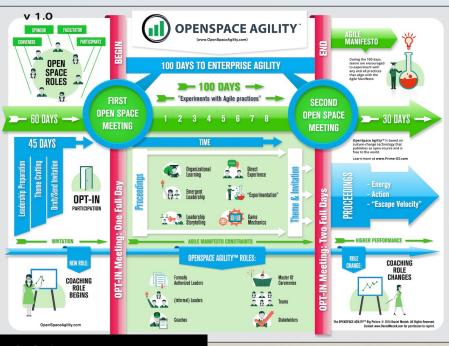
agendashift.com

Open Space Technology / Open Space Agility



Open Space Technology / Open Space Agility

"OpenSpace Agility is a safe, pragmatic and repeatable technique for getting a rapid and lasting Agile adoption. It works with the framework you are currently using..."

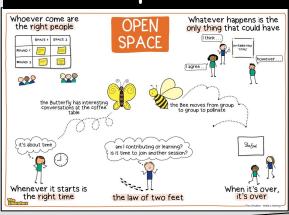


- 1. Leadership and Enterprise Preparation
- 2. Initiate the process using an all-hands "Open Space" meeting
- 3. Initiate Agile practices across the enterprise
- 4. Complete the process in Open Space
- 5. Inspect Results and Adapt

OSA Steps

OST is simple & effective Self organizational OSA focus on invitational agile transformation Aligned with Agile Manifesto

OST Principles & Laws



be prepared to
transfer change
to the group
OSA potentially
disruptive
OSA adoption small
but growing

OSA Big Picture









Holocracy / Sociocracy



Holocracy / Sociocracy

"...Radically changes how an organization is structured, how decisions are made, and how power is distributed."

Circle Structure

Board Company Circle Circle Development Department Operations

Organizational Structure Rules of Cooperation **Tactical Meetings Distributed Authority** Governance Process

Consent Effectiveness Continuous Improvement

Holacracy Constitution



Execution clears space for new

changing DOING IT

Clarity allows sensing tension

No clear voice of customer May lessen transparency the further from the circle you are Full organizational cultural change required Limited uptake

Operations

Work Flow

projects

Leverages

Sensing

Increases

organizational capacity

Structure hared author

Governance

Reality provides ➤ insight

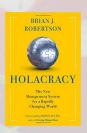
meetings for

drives nchronization

drives integration

Purpose

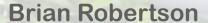




Encourages team members to take initiative Distributed authority & decision making

Sociocracy

Principles



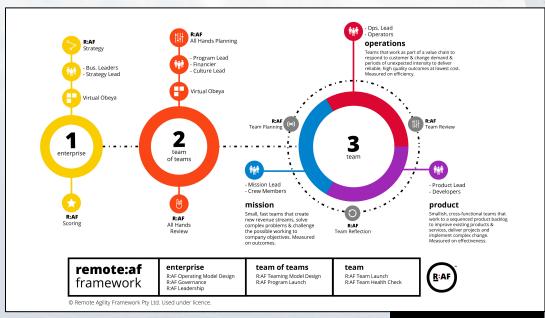


holacracy.org / sociocracy30.org



Remote Agility Framework

"...helps organisations to build remote and hybrid workplaces that deliver amazing results and keep staff engaged and motivated."



Principles

Trust in People Strategy Evolves Clarity is King Remote but responsible

All Teams are Equals
Pass things Gently.
Help Each Other.
Measure what
counts

Respect Circumstance Work smaller Further, but Closer Tool the F up

Remote first
Modular & tool agnostic
Replicates physical
approaches virtually

Framework



Andrew Blain
David McKenzie
Natalie Valcuikas
Paul Fitzmaurice

Nikola Peter John Tooth James McMenamin Phil Gadzinski David Martin Tony Ponton
Denton Currin
David Berry
Farnaz Vahab

Contributors

Toolkits need certification or community access
Build around remote toolsets
New model, adoption low, limited documentation

Cynefin

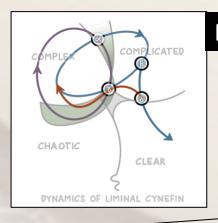


Cynefin

"...at its heart a decision support framework that distinguishes order from complexity from chaos"

Model





Dynamics

Continually updated
Well regarded
Lots of methods & tools
supporting the model

Can be hard to understand & apply

Cost to access in depth material & Sensemaker software



Dave Snowden 1999 thecynefin.co



Oath of Non Allegiance



Oath of Non Allegiance





I promise not to exclude from consideration any idea based on its source, but to consider ideas across schools and heritages in order to find the ones that best suit the current situation.





Thank You



Craig Smith

Osmithedau

craigs Osofted.com

craig Ocraig smith.id.au





2022 Edition

