QIpyuS pa'Daq jIbuS

QumwI'pu' Human sutlu', 'ach vongwI'pu' 'uy' ghe'naQmey qengbogh toy'wI', ghe''naQmey le' ghaH . pay' jonpIn wa'DIch je, pay' ghu' je. naDevvo' yIncho H 'ej naDqu' muSHa'ghachchaj, jIQublaHbe' 'e' vI'Iprup.

Creators

Jeff Sutherland John Coleman Ralph Jocham

2025-06-15

Contents

0.1 mo’ ’elmeH . . . 3
0.2 QIpyuS pa’Daq jIbuS ngoQ . . . 3
0.3 QIpyuS ghom vIghro’ rur . . . 3
0.4 QIpyuS boQ je buSHa’meH QeD . . . 4
0.4.1 Qapchu’be’ ghu’mey QIpyuS’e’ buSHa’bogh {#complexity–the-case-for-scrum} . . . 4
0.4.2 narghghach . . . 5
0.4.3 QIpyuS ghom Qorghwl’ le’ . . . 5
0.4.4 laHlIj . . . 5
0.4.5 nItebQeD . . . 5
0.4.6 Qubchu’ghach . . . 5
0.4.7 tlhegh . . . 5
0.5 QIpyuS Qubchu’ghach Qapmey wej {#the-three-pillars-of-scrum’s-empirical-process-control} 6
0.5.1 leSpoH . . . 6
0.5.2 QeD . . . 6
0.5.3 choH . . . 6
0.6 The Scrum Values . . . 6
0.7 More Supporting and Complementary Theory . . . 8
0.7.1 Product Thinking . . . 8
0.7.2 Systems Thinking . . . 9
0.7.3 Discovery . . . 9
0.7.4 Leadership . . . 9
0.7.5 First Principles Thinking . . . 10
0.7.6 People and Change . . . 10
0.8 The Scrum Roles in the Expansion Pack . . . 11
0.8.1 Stakeholder . . . 12
0.8.2 Supporter . . . 14
0.8.3 Artificial Intelligence . . . 14
0.8.4 Product Developer . . . 15
0.8.5 Product Owner . . . 16
0.8.6 Scrum Master . . . 17
0.9 The Scrum Artifacts in the Expansion Pack . . . 19
0.9.1 Product . . . 20
0.9.2 Increment . . . 21
0.9.3 Product Backlog . . . 22
0.9.4 Sprint Backlog . . . 24
0.10 The Scrum Events in the Expansion Pack . . . 24
0.10.1 The Sprint . . . 25
0.10.2 Sprint Planning . . . 26
0.10.3 Daily Scrum . . . 26
0.10.4 Sprint Review . . . 27
0.10.5 Sprint Retrospective . . . 27

0.11 Multi-Scrum-Team Products . . . 28
0.12 End Note . . . 30
0.12.1 Acknowledgments . . . 30

1 **30**
1.1 Scrum *Expanded* on One Page . . . 30

2 **32**
2.1 Expansion Log . . . 32
2.2 Appendix . . . 33
2.2.1 The Adaptive Enterprise . . . 34
2.2.2 The Adaptive Executive or Board Member . . . 35
2.2.3 Cynefin® {#cynefin®} . . . 36
2.2.4 Emergent Strategy . . . 40

3 **42**
3.1 Attribution for the Scrum Guide Expansion Pack Collection . . . 42

4 **42**
4.1 References . . . 42

ngoq’a’ Scrum Guide chenmoHpu’bogh Ken Schwaber je Jeff Sutherland (40) wIqel

Scrum Guide Expansion Pack boQmey luwIvlu’bogh

ngoqvamDaq De’ luwIvlu’bogh ’oH, ’ej pabmeyvam HochHom tIbmey luwIv. Sep Hoch vIqelmeH, pab lugh DalaHmeH, lut tIlaH.

wonmoH 1: Scrum Guide Expansion Pack wa’ (lIwghach)

pong: Scrum Guide Expansion Pack lIwghach ngoq’a’: The original Scrum Guide

qonta’: Ralph Jocham, John Coleman, Jeff Sutherland

’och: 2020 Scrum Guide, Scrum Guide Expansion Pack

chut: Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

© 2025 Ralph Jocham, John Coleman, Jeff Sutherland

chenmoH le’: ngoqvam chu’ veblu’meH 2020 Scrum Guide lIwlu’. choHmey tu’lu’.

qapbe’lu’chugh, bIje’laHbe’. bIvangtaHvIS, SoHvaD qonlu’.

ngoqvam bopbogh pab CC BY-SA 4.0 tIchup. bIlo’chugh, bIHoHQo’laH ’ej bIlabqangmeH, pabvam yIlaj.

0.1 mo’ ’elmeH

Ken Schwaber je Jeff Sutherland QIpyuS Qun chenmoH Devpu’. 2020 Scrum Guide (40) QIpyuS potlhqu’ ‘oH. Tobias Mayer’s A Simple Guide to Scrum (58) ’oH ngoq’a’ ghomwI’ nIvqu’ QIpyuS chenmoHbogh Ken Schwaber je Jeff Sutherland. Scrum Hexis (52) 2025 wa’DIch ngoq’a’ veQmey chu’ ‘oH. QIpyuS ’ay’vam nIvbogh ’oHbe’ ‘ach tagha’ Qapla’meH, QIpyuS ngoQ vInIS: nap ’oH.

0.2 QIpyuS pa’Daq jIbuS ngoQ

QIpyuS pa’Daq jIbuS chu’wI’ Qapchu’meH, 2020 Scrum Guide (40) naQ ‘oH mIw chu’moHmeH net lo’. Ralph Jocham (89) ‘oH ’ay’vam Qav mIw chu’ neH ’oH.

QIpyuS pa’Daq jIbuS ‘oH QIpyuS ’ay’mey Hoch *nuq* je *qatlh* DelmeH. Hoch ’ay’ motlh ngoQ ghaj, je QIpyuS QapmeH, lo’laHghach je chenmoH. QIpyuS pa’Daq jIbuS taHtaH. wa’DIch poH Hochmey laHIIj DalaH.

ngoqvam laH QIpyuS HoI yIqeq. QIpyuS 2020 vIlaHchu’pa’ ngoqvam yIlaHchu’. pongmey ‘oH bopbogh maqoch vIt. Appendix je pongmey boQmey yItu’.

QeDpInpu’, ghomwI’ je QIpyuS yIlo’meH, neH Daj, nom, SuvwI’, ghojmoHwI’, nabwI’, nIvbogh, QeD, SIQpu’, ‘ej Qav Qapla’ neH. Qub ‘e’ wIneH: QIpyuS yIlaHmeH, ghaHDaq Qav QeD Hoch lo’taH—QeD, QuvwI’, De’ je Hoch Hom.

QIpyuS pa’Daq jIbuS QapmeH Hoch QapmeH QulmeH, QulwI’, QulbogH, QulwI’ QulmeH je ghom QIpyuS neH ghomvam. ‘e’ vItbe’chu’, QIpyuS laH tlhoS Hoch DaqmeyDaq, Qapla’ wIbuSmeH Hoch Qu’mey vIchenmoHlaH. ngoqvamDaq QIpyuS vIlo’chugh, QIpyuS ‘oH Qav Qapla’ meH Qul.

ghomwI’ lo’laHghach ‘oHbogh, ghomvam chenmoHbogh, tlhoblaHbogh (customers, users, decision-makers je) Hoch QelbogH lo’wI’ QaptaH. ‘ach ghomwI’ Hoch SovlaHbe’. mIw, ‘ej tu’lu’ ‘e’ vItlu’. vIt Sovmo’ SuvrupnISlu’, ‘ej vIt SovQo’ QaptaHvIS, vIt SovnISlu’. vIt wovbe’ Qapbe’nISlu’, ‘ach Qav Sov je rurlu’.

0.3 QIpyuS ghom vIghro’ rur

QIpyuS ‘oH mIw’e’ malja’ QulchuʼbogH, ghot Sov neH not yap, ‘ej ngoD qelDI’ poH juS, rurchuq ‘oH. QIpyuS Hoch malja’ yInDaj vItlhutlh: chenmoH, choH, pol, leH, SIp, QI’tu’, bup, Qav je. QIpyuS ghotpu’, ghom, ’ej

yejmeH mIw leH, choH QorghmeH, lo’laHghach chenmoHmeH.

QIpyuS QorghmeH ghomwI’ Sovchu’, vaj ‘oHvaD lo’laHbe’ Sov Qorgh. QIpyuS meqmey chu’ je ‘angmeH, narghmoHghach SIQbe’meH, je SIQchoHlu’meH Qapla’ vIneH. ghom Qorgh ‘e’ QaH QIpyuS, Qapchu’choH.

Qob ‘oH Dochmeyvetlh’e’ ngaSbogh ngoDmey, vaj ‘oHvaD wanI’mey QaD vIlo’. Qob ‘oH ’ach leSpoHmeH, pagh Sovlu’. Qob chuq laH Qorgh QIpyuS: malja’ Qob, De’ HaqlaHbe’, pat choHbe’chu’, SIgh, Hevwl’ chutmey, lughbe’ghach SeH, je. QIpyuS lo’taHvIS, Qob SuvwI’pu’ tIbachHa’.

QIpyuS QorghwI’ je SuvwI’ Daq ghorgh SIQ, lugh Qapla’ luwIv je.

ngoqvam vItlhutlh:

1. QorghwI’pu’, vaj Supporters ponglu’, QIpyuS QorghmeH Qapchu’ QutlhnIS, QIpyuS DevwI’vaD SuvwI’ Qorgh je.
2. malja’ laHchugh QorghwI’, malja’ ngoQ wIv.
3. QIpyuS ghom le’ (49), Qu’mey wIv, QIj, je lo’laHghach chenmoH.
4. QIpyuS ghom je QorghwI’ DuHmey SuvmeH SprintDaq ghu’ SIQ je choH.
5. Supporters QIpyuS ghom Qapchu’meH boQ.
6. *chegh.*

HInob: chu’ veb pagh QIpyuS QorghwI’vam wIv, ‘ej QorghwI’ vItlhutlh je. malja’ Qap ‘oH je. vaj ’oHvaD lI’ wIv wIvlu’taH QIpyuS.

QIpyuS ‘oH veb QIghbe’. QIpyuS chutmey ‘oHbe’. ‘oH rIQ pagh ghom SeHlu’, ‘ach Qapla’ QorghwI’ je qonlaHlu’.

QIpyuS lo’chugh: QapwI’mey chu’ yIqel, je DevwI’ ghom ‘e’ DaSuq.

rugby Qujvo’ pongwI’ ‘oH “Scrum” pong’e’. Takeuchi je Nonaka (29) QIpyuS ghomwI’ chu’ vItlhutlh, pa’ Sov chIw’e’ rIH, ‘ej malja’ le’ chenmoHlaH.

0.4 QIpyuS boQ je buSHa’meH QeD

QIpyuS QaptaH ghom QorghwI’ le’, narghghach, Qubchu’ghach je ‘IQ. buSHa’meH QeD ’ach boQchu’ QeD DuQbogh:

- ‘umwI’ pIch
- nIDHa’meH meqmey tInbe’ (yejquv ’oHbe’meH)
- Qap bIquv rur nabmey
- tu’ghach, HInob, lo’laHghach ’ang
- taHtaHghach

0.4.1 Qapchu’be’ ghu’mey QIpyuS’e’ buSHa’bogh {#complexity-the-case-for-scrum}

malja’ QulmeH Qu’meyDaq, Sovbe’bogh ‘oH ’Iq. Sov law’ Sov puS, ‘ach yapbe’. QeDvam buSHa’bogh (30-35) QubwI’ lI’ je ‘ang. QIpyuS ghomwI’ poH poQ: Qub, Qochbe’ghach, QI’lo’, Qubqa’. Qub rurchuq je Qubchu’ghach lI’.

Sovlu’bogh Hoch, malja’ je QorghwI’ (customers je), lIwchu’laHbe’. Qubchu’ghach lI’ lo’lu’chugh, nIDmey buSHa’bogh, SIQ je choHmey Del.

wanI’ rapDaq QaptaHvIS QIpyuS ghomwI’, Qubqu’ ghu’. ‘op poH, Qubmey lo’laHbe’qu’, ‘ach Qubqa’ ghaj. ‘op poH, QIpyuS ghom wIv wIbuS, SuvwI’ DuQ, ghIq Qaplaw’.

Qub rurchuq ‘oH: nuq, qatlh, chay’, ghorgh, nuqDaq, ‘Iv. QaptaHvIS yIlo’: loQ yIDach, yIvang. QIt yInIDQo’.

bItrust’e’ chenmoH ‘e’ vItlhutlh: ghom QorghwI’ le’, yI’oyQo’, yIchoH, QeD chu’, chu’ Sov.

0.4.2 narghghach

narghghach (71) ‘oH ghu’vam’e’: SuvwI’ vaQ ‘oH HoS ghaj. QIpyuS Qap, cha’logh: narghghach SeHbe’, ‘ach poQwI’ lupoQ: poH SeHwI’, QuvwI’, jangwI’ yItlhutlh. QeDvam vIll’: ghom QorghwI’ chenmoH, SuvwI’ qeng.

ghom QorghwI’ Qapchu’ ‘e’ wInID: Qub, QI’lo’, nIDmey. chut ll’: *Swarming*, *Stable Teams*, *Kaizen*. QIpyuS ghomwI’ Qubmey’e’ QubchoH. ghaH Qubmeymey ‘ang je: chu’ mIw Qapla’ ‘oH. Qubmey Qorgh, bIje’, yIll’.

QorghwI’ le’ (49) chutmey Dalo’chugh, QIpyuS ghom Qapqu’ je. chut nIv ’oH, QIpyuS Qapqu’meH Qubmey chenmoHmeH.

0.4.3 QIpyuS ghom QorghwI’ le’

QIpyuS ghom QorghwI’ le’ (49) QaptaH: QaplaH’a’? QaplaHbe’chugh, choH. Qu’ nuq vIvang? QIpyuS ghom QuvwI’ mIw QaplaH. QapHa’ ghu’ luwIvchugh QorghwI’, ghom QorghwI’ DapeghQo’.

ghom QorghwI’ le’ malja’ ghajchugh, QIpyuS Qapchu’. ghom QorghwI’ le’ Qap, ‘ach ghot QorghwI’ le’ Do’Ha’ Qapbe’. QeDvam QIpyuS QapmeH: chut laH Qorgh, QIpyuS QaplaHmeH.

0.4.4 laHllj

laHllj ‘oH: tIvchu’ Qap, yItIv je, Qapchu’. Quch je. laHllj ‘oH: Qap vItlhutlh. QapHa’ laHlljbe’.

laHllj QIpyuS ghom QorghwI’ vItlhutlh: ‘ogh, pol, ll’, ghojmoH. ngoq ll’: Specification by Example, Clean Code, Unit Testing, TDD, CI, CD, Architecture, Design, Acceptance Testing, Testing je QaplaHghach.

0.4.5 nItebQeD

nItebQeD (63): QobHa’ Qap, QaplaHghach ghaj, lo’laHghach ll’, taHtaHghach. QIpyuS QaplaH, nItebQeD ll’. nIDtaHvIS, QaptaHvIS je. nItebQeD Qapchu’, malja’ Qapchu’meH.

0.4.6 Qubchu’ghach

Qubchu’ghach (67): Qub ghajmeH, De’ laHbe’be’ vItlhutlh. QIpyuS QaptaHvIS Qub. ghot Qub: Sovlu’be’. Qubchu’ghach Qap: QeD, QubchoH, SIQ, ll’, QaplaHghach.

QIpyuS QIgh: Qub wIv, Qub lo’, Qap ‘e’ yItlhutlh. Qubchu’ghach chu’ ’oH. Qubmey: trade-offs, nIDmey. nID: Qub laHlu’bogh. Qubmey vInej. nID vItlhutlh.

Qubchu’ghach je nItebQeD buSHa’ghach chu’ ‘oH: QeDvam yIlo’.

QIpyuS QapmeH, QorghwI’ lo’laHbe’ Qub wInej. Qubmey DeI, Qap. Qapchu’ghach SuvwI’ vItlhutlh.

0.4.7 tIhegh

SprintDaq Qap: ghom QorghwI’ le’ wIv. tIhegh ‘oH QapmeH, Qapchu’. ghItlhmey, Qub, ll’, Qubchu’ghach, vItlhutlh.

0.5 QIpyuS Qubchu'ghach Qapmey wej {#the-three-pillars-of-scrum's-empirical-process-control}

Qubchu'ghach: Sov wIvlu'. Sov: SIQ, Qub. QeDvam 'oH: Qub Qap (64-66). Qubvam QIpyuS Qap: Transparency, Inspection, Adaptation.

0.5.1 leSpoH

leSpoH 'oH QIpyuS Qap. QeD, Qubchu'. mIw, Qu', QeD, QaplaH, Hoch vItlhutlh. QeDvam 'oH. 'oHvaD Qap: Product Backlog Items, Increments.

QIpyuS QapmeH, De', nID, QIgh, Qap. leSpoH QapmeH, yIlegh.

De' Qap: Qub lo'laHghach, QuvwI', meq, nID.

leSpoH Qapbe'chugh, QIpyuS bIQapbe'.

0.5.2 QeD

QeD QIpyuS Qap. Qub, QeD Qub, Product Goal. QeD: Qub. QeD: Qub QeD. poH: Events.

Artifacts, goals, QeD. De' je. QeD: leSpoH, Qub, QIgh.

0.5.3 choH

choH Qap. Product QeD, Qub, QaptaHghach Qub. Qapchu'.

choH taghmeH: Qub yIlaj. QeD DalaH: Qub wIv, Artifacts, ghom QorghwI', QorghwI' je. De' QavlaHbe', Qub yIchoH.

leSpoH, QeD, choHbe'chugh, QIpyuS Qapbe'.

0.6 The Scrum Values

The Scrum Values —*focus*, *openness*, *commitment*, *courage*, and *respect*—help create a Scrum Team environment that supports psychological safety and positive collaboration, which align with principles identified in neuroscience as beneficial for learning and effective teamwork. Consider the context.

The Scrum Values foster transparency and trust, ensuring that words and actions align. Together, they create a strong foundation for collaboration, performance, and coherence within a Scrum Team.

Successful Scrum adoption depends on the Scrum Team and Supporters (and other Stakeholders) leading by example as professionals. The Scrum Values can help improve trust among the Scrum Team and Stakeholders. The Scrum Values also encourage ethics (57), vocabulary, tone, work, behavior, and action that foster trust. They also help reduce or avoid a gap between words and actions.

The Scrum Team and Supporters agree to be *open* about all the work and the challenges. Humility supports *Openness*. *Openness* requires trust, and trust requires *Openness*. The Scrum Team and Supporters should request and share constructive feedback. They routinely collaborate and learn through high-bandwidth conversations and qualitative or quantitative feedback.

High-bandwidth conversations are conversations that foster communication in ways that allow for the richest, fastest, and clearest exchange of information. This typically involves face-to-face discussions-either in person, via video calls, visual management, or whiteboards (physical or digital), where participants

can use not just words, but also tone of voice, facial expressions, drawing, or body language to fully understand each other.

As Sprints are short, any failures should be small and quick, and risk is identified and managed through fast and *open* feedback. Perhaps, the only real failure is the lack of learning.

The Scrum Team and Supporters should have the *Courage* to do the right thing and face tough challenges. They should be courageous in exploring the unknown, changing direction, requesting and sharing information, and engaging in courteous disagreements, e.g., healthy conflict and constructive dissent. The Scrum Team should ask Supporters and leaders for help if needed.

The Scrum Team *Commits* to achieving the Sprint Goal and supporting each other. Commitment means getting relevant work toward the Sprint Goal to comply with the Definition of Output Done at the latest by the end of the Sprint, preferably much earlier. Commitment also means getting to desired outcomes through value realization.

Their primary *Focus* is making the best possible progress toward the Sprint Goal. Their secondary Focus is making the best possible progress toward the Product Goal. The Supporters *Commit* to providing a psychologically safe space and environment for the Scrum Team to deliver Increments; within their *Focus*, the Scrum Team and Supporters *Commit* to creating time for continuous learning and Adaptation, and the transfer of learning between Scrum Teams to ensure long-term effectiveness. The Scrum Team and Stakeholders should be intentional about addressing trade-offs, including consideration of short-term wins with long-term consequences.

The Scrum Team and Supporters (and other Stakeholders) *Respect* each other as skilled professionals; they *Respect* each other's differing expertise and perspectives and are constructive when disagreeing. Respectful behavior supports trust. The Scrum Team and Supporters should critique the idea or the approach to find more effective options, not the person(s).

Respect helps protect against the weaponization of the other Scrum Values. Demonstrations of *Respect* can include (but are not limited to) genuine praise, support for each other, humility, psychological safety, constructive disagreement, and cognitive diversity.

Scrum Team members and Stakeholders can look at the Scrum Values through the lens of John Boyd's OODA (99, 100, 102). OODA was created by U.S. Air Force Colonel John Boyd to help pilots make quick, smart decisions in fast-changing situations by moving through four steps: Observe, Orient, Decide, and Act. It's a simple, continuous, iterative, powerful, if often subconscious way to handle uncertainty—like noticing market changes (Observe), analyzing trends and risks (Orient), choosing Product features to test (Decide), and delivering them (Act). OODA helps individuals stay flexible and respond well to whatever comes their way. Scrum can improve OODA.

Individual Scrum Team members can look at the Scrum Values through the lens of John Boyd's OODA, using Scrum to foster emergent solutions. In a Scrum context, the Scrum Values apply across all of OODA, and in particular, help as follows:

- Observe—Openness and Respect can foster the gathering of all relevant evidence, and diverse perspectives.
- Orient—Courage is needed to interpret reality, navigate uncertainty, and agree to adapt or pivot, potentially using a reflective pause to challenge assumptions and provoke new insights.
- Decide—Deciding what to do requires timely analysis, such as backlog refinement, bringing potential next steps into Focus through parallel safe-to-fail experiments to test hypotheses, like small-scale

probes (probes should be small, parallel, and designed so that failure is survivable and informative).

- Act—With clarity on what needs to be done, why, and by whom, Commitment can drive the team to execute effectively within enabling constraints like time-boxed sprints, fostering emergent solutions.

0.7 More Supporting and Complementary Theory

0.7.1 Product Thinking

People consume Products (including services), not projects. A Product is the conduit to deliver value, balancing the short- and long-term. This is why Scrum has a Product Owner and not a Project Owner. Products are long-term and need to be taken care of for their entire existence, whereas a project is time-boxed and often leaves an orphaned Product behind once the project is completed.

Product thinking (86-88) deals with the tension (111) that Products often need to *Focus* on growth in the short-term but also need to address long-term concerns, e.g., gaining traction with early adopters, 'crossing the chasm' (5), expanding, updating Product versions, continuous change, customer lifetime value and total cost of ownership.

To 'cross the chasm,' a strategy shift is needed from targeting savvy, risk-taking customers to winning over more pragmatic, risk-averse buyers, decision-makers, users, or other Stakeholders by focusing on a specific niche market or target and delivering a complete, reliable solution that solves real problems. This step is crucial for a Product's transition from niche success to widespread adoption, as it moves from appealing to early adopters to attracting the early majority. The early majority often requires clear evidence of the Product's reliability and problem-solving capabilities within a specific context. By focusing on a niche and providing a complete solution, a company can build credibility, create reference customers, and establish a strong market position, effectively bridging the 'chasm' between early adopters and the mainstream market.

Product Owners need to master the handling of trade-offs between the *here* and *now* and the anticipated future (the *there* and *then*) (148) through courage, humility, consultation, collaboration, healthy conflict, etc.

Suppose the people involved are purely short-term thinkers. In that case, they will likely experience long-term side effects such as technical debt, low Scrum Team morale, busyness, output focus, etc. For that reason, mitigating factors would need to be put in place to support the long-term.

Technical debt is the extra work that builds up – consciously or unconsciously – when you take shortcuts in implementation or design to deliver something faster. Over time, it slows you down, just like real debt—with interest—because it makes future changes harder and riskier. Professionals strive to minimize technical debt and sloppiness as much as possible. If they decide to incur debt, it should be transparent, and if possible, an emergent mitigation plan should be in place.

For Products, Scrum supports feasibility, usability, desirability, value, and viability within ethical (57) boundaries through:

- Product design
- Product management
- Intentional consideration of the cohesive interplay of Stakeholders, research, goals, discovery, design, delivery, and continuous value realization
- In the specific case of technology Products, through Product engineering.

Scrum favors a healthy balance of the short-term and the long-term. Goal orientation enables potential outcomes through an emphasis on value and risk reduction. The Sprint Goal *(here* and *now)* should be a step toward the Product Goal *(there* and *then)*, which enables pathways to the long-term. The Product Goal often supports the Product strategy and Product Vision.

0.7.2 Systems Thinking

Systems thinking (55) acknowledges the interconnectedness of elements within organizational and social contexts, recognizing that actions in one area ripple in ways that aren't always predictable or linear. Theory-informed experiments, feedback loops, and follow-up data analysis help surface valuable and actionable insights. Systems Thinking provides valuable tools and ideas and facilitates insights.

For an organization to become adaptive (80), it is necessary to avoid local sub-optimizations such as reducing unit costs while increasing long-term costs, eroding quality goals only to lose customer trust, or improving a Scrum Team, workflow, or process that should not exist. For complex work (30-35), it's not always possible to link cause and effect, except in hindsight. It's helpful, nevertheless, to consider possible and actual upstream, cross-stream, and downstream effects of interventions.

0.7.3 Discovery

Discovery (50-51) often starts with understanding people's expectations, needs, and wants through observation, analysis, conversations, and synthesis toward a desired outcome. Once a Scrum Team has gathered insights, it frames the problem or opportunity and orders them by potential value. The Scrum Team crowdsources possible solutions without judging them too quickly. If the potential value is high but there is a lack of evidence that the value can be realized, the Scrum Team should do research, assumption testing, or build simple prototypes they can test with real customers, decision-makers, or users. Discovery is never over; consider regular interviews or observations of customers, decision-makers, or users.

Discovery is about learning toward a desired outcome through prioritizing, doing, avoiding, or constantly improving ideas informed by user observation, feedback, or other learnings. Discovery emphasizes collaboration, creativity, and not being afraid to fail and try again. Discovery frames work as problems or opportunities and helps the Scrum Team to create, prioritize, and test solution options that balance what people desire, what's technically possible, and what makes business sense—all while having fun.

If discovery is needed, it should (insofar as it is possible) be included in a manner that is consistent with Scrum. For example, discovery work is made transparent in the Product Backlog and Sprint Backlog, Scrum Team members practice discovery and other skills, learnings are discussed during the Sprint and at the Scrum events, and at least one Increment is produced (and ideally released) every Sprint, regardless of how much discovery is done. There is a balance to be struck: discovery can help avoid building the wrong thing, but it can be overdone, and, in the end, the result feedback matters the most.

0.7.4 Leadership

Leadership is the ability to influence, guide, and inspire a group of people to achieve a common goal while avoiding demotivation. It inspires thoughts, actions, and passion and fosters clear strategic directions. It embraces purposeful and intentional Go See, Listen, and Understand, collecting facts and observations to inform decisions, better known as Genchi Genbutsu (84).

Leadership is a dynamic social process involving responsibility, relationship building, and empowerment. Successful leadership results in co-creating a direction of travel, effective alignment of resources and people needed, and mutual *Commitment* among group members.

Scrum strives for a particular leadership, that is, leadership for resilience, a set of qualities, not a management position. Thus, leadership needs to include but not be limited to cultivating the environment for self-managing Scrum Teams, clarity, trust, transparency, emergence (71) in a direction, fulfillment at work, embracing uncertainty (72) and failures, gathering evidence for better decisions, proactively managing risk, and removing organizational inefficiencies.

Leadership happens from all angles, should be at all levels, and fosters reflection at the right times. Leadership should drive ruthlessly for value, yet be compassionate and ethical. Leadership requires persistent agency to change workflows, processes, systems, and the work environment; this includes (but is not limited to) HR, finance, and vendor management. A leader is someone who demonstrates leadership.

Product Owners and Scrum Masters balance leadership, authority, and subtle control by providing clear intent, fostering initiative, and reinforcing accountability. They guide rather than micromanage, ensuring the Scrum Team understands the vision and goals, has the autonomy to execute, and remains accountable for outcomes. When intervention is needed, they step in decisively while preserving the Scrum Team's ownership of their accountabilities. Product Developers demonstrate leadership with their self-managing team orientation, professionalism, and goal orientation; self-management comes with responsibilities. Supporters demonstrate leadership by supporting short- and long-term impediment removal, improving the coherence of management processes with Scrum, and supporting emergent change in a powerful direction when requested.

0.7.5 First Principles Thinking

First principles thinking is a method of problem-solving that involves breaking down challenges into their most fundamental truths and discovering solutions from the ground up. Instead of relying on analogy or established conventions, this approach asks, *'What do we know for certain?'* and reconstructs understanding and solutions from those basic elements. Examples could include but are not limited to:

- Encouraging the Scrum Team to *Focus* on the core drivers of effectiveness, adaptiveness (80), and timeliness -such as autonomy, transparency, and adaptation- rather than blindly following processes or copying what others have done.
- Questioning every assumption and reconstructing solutions based on facts and essential principles, which can enable breakthroughs.
- Advocating original thinking, continuous improvement, and the *Courage* to challenge the status quo-unlocking creativity and enabling transformative results.

0.7.6 People and Change

The level of difficulty in adopting Scrum should not be underestimated. Scrum offers some guiding principles through its elements. It offers an approach to go back to first principles.

Scrum is not about adopting tools. And Scrum doesn't end with impediment removal. An impediment in Scrum is anything that blocks or slows down progress. It is crucial to be intentional, unrelenting, and

tenacious about people, change, and communications. The change often includes people development, designs, workflows, processes, systems, attitudes, behaviors, language, habits, and the work climate. Culture is an emerging result.

An effective Scrum adoption uses an emergent approach, has effective change agents, and engages enthusiastic support from those affected by it or affecting it. Intentionality and daily progress with the adoption are crucial; the adoption work should not be the last thing that is worked on after everything else is finished.

Start with disciplined emergent change in a direction. Strive to make emergent change so normal that it eventually becomes part of the scheduled work. A Scrum adoption has direction but not a predefined destination. The change is emergent and therefore not predictable. Curiosity enables a pattern of sense, listen, learn, and adapt in a direction. It's important to foster relationships and understand perspectives, and to listen to what is not being said and what is not happening. Change is hard work, yet fulfilling.

0.8 The Scrum Roles in the Expansion Pack

The four Scrum roles are Product Owner, Product Developer, Scrum Master, and Stakeholder. They give, reward, and earn trust and enable coherent leadership. Only the three accountabilities, Product Owner, Product Developer, and Scrum Master, are in the Scrum Team.

A person can hold more than one Scrum role. By taking on more than one role, one must be careful not to overstep. The Scrum roles are designed to keep check and balances in place.

A Scrum Team is a team that practices Scrum, addresses three Scrum *accountabilities,* namely, Scrum Master, Product Owner, and Product Developers, deals with Stakeholder (including but not limited to customer or user) problems or opportunities, and delivers useful, usable, and potentially valuable Increments from the perspectives of the Scrum Team and Stakeholders toward the Product Goal. For complex (30-35) work, a Scrum Team should be small, cognitively diverse, and self-managing, where human Scrum Team members, often assisted by technology, care about each other's work and learn how to do each other's work.

The Scrum Team should be cross-functional, which means it is multidisciplinary, including technical and business domain skills. There is no explicit hierarchy within the Scrum Team. The Scrum Team should have all the skills and support needed to:

- Discover (including research and design) as needed,
- Deliver (including engineering when appropriate); and,
- Validate value realization (and additionally usability, desirability, and viability within ethical (57) boundaries).

The Scrum Team, supported by the Supporters, collectively takes care of the problem or opportunity domain, Product discovery, delivery, verification and built-in quality, go-to-market, and value validation toward the Product Goal. The Scrum Team strives for net improvements; being self-managing (49), they decide *who* does *what*, *how*, *when,* and *where*.

Value validation is the confirmation (or disconfirmation) within given boundaries that the expected outcome(s) has been achieved.

The Scrum Team delivers an Increment(s) every Sprint, continuously self-manages (49) to find and fix problems, synchronizes continuously, and releases frequently. The Scrum Team is small enough to

remain nimble and large enough to complete significant work within a Sprint. Often, smaller Scrum Teams communicate better and are more productive.

Scrum is built on self-managing Scrum Teams (49) within a defined organizational or Product structure. Autonomy exists, but it is bounded by Scrum events, accountabilities, artifacts, commitments, pillars, values, and organizational needs.

Scrum engages groups of people who collectively have or acquire all the skills and expertise to do the work and share such skills as needed. Intentional interactions, supported by leaders, are needed to help improve the chances of successful outcomes.

The *Focus* should remain on achieving the Product Goal in the most effective way, with the right amount of investment, while delivering valuable outcomes.

Scrum fosters collaborative teamwork by encouraging continuous interaction and shared accountability rather than sequential, siloed work. This approach enables the Scrum Team and Stakeholders to embrace uncertainty (72), allowing for quicker adjustments informed by ongoing learning and feedback. The overlapping nature of discovery, development, and value validation ensures a more adaptive (80), value-driven approach to Product development.

Overlapping work fosters shared accountability among the Scrum Team, enhancing engagement and commitment. The Scrum Team directs attention to addressing challenges and opportunities, encourages proactive behavior, cultivates a diverse skill set, and increases awareness of market dynamics among all participants.

The Scrum Team addresses all Product-related activities, from Stakeholder collaboration to value validation, including verification, maintenance, operation, experimentation, research and development, and anything else that might be required. The Scrum Team instills quality. Supporters ensure the organization structures the climate and the work environment and empowers the Scrum Team to self-manage (49). Working in Sprints at a sustainable pace improves *Focus* and consistency.

The Scrum Team and Stakeholders don't know what they will learn. Some learning provides greater certainty, and some creates more uncertainty (72). Some things could emerge, fade away, drop out, or become less of a priority.

A Scrum Team has aligned autonomy. Aligned autonomy means the Scrum Team has the freedom to decide how to solve problems while staying focused on shared goals and outcomes, enabling both innovation and organizational coherence. Fostering a cognitively diverse Scrum Team is essential. Cross-pollination is more likely when the Scrum Team members collaborate, trust each other, and can self-reflect.

For successful outcomes, the Scrum Team and Supporters should cultivate a willingness to unlearn outdated principles. Inspection and Adaptation require a climate that anticipates and tolerates mistakes. It is essential to *Focus* criticism on ideas rather than individuals. All Scrum Team members 'play on the same field' with coherently overlapping work, and are all accountable for success.

0.8.1 Stakeholder

Stakeholder is a role. A Stakeholder is an entity, individual, or group interested in, affected by, or impacting inputs, activities, and outcomes. Stakeholders have a direct or indirect interest inside or outside the organization, its Products, or services.

Examples of Stakeholders include (but are not limited to) customers, decision-makers, users, vendors, influencers, managers, colleagues, leaders, legislators, financial sponsors, subject matter experts, and

governance oversight. Inanimate, non-human Stakeholders such as the law or AI should not be ignored. Some Stakeholders have more impact or are more impacted than others, and each can favor different factors. Each Stakeholder has a different degree of power or influence.

A customer is any Stakeholder who receives value from the Product by purchasing and/or selecting it. Customers may include buyers (those who pay for or acquire the Product), decision-makers (those who approve or authorize its adoption), and end users (those who interact directly with the Product). Sometimes, the customer is not the same as the end-customer, e.g., in B2B2C (79) or B2B2B (78).

For a successful Scrum adoption, it's crucial to have regular intentional interactions between Stakeholders (including but not limited to customers and users) and the Scrum Team.

A user is a Stakeholder who directly interacts with the Product to achieve specific goals or solve problems. Users experience the Product firsthand, whether it is a service, platform, or experience, and their feedback and satisfaction are crucial for ongoing Product improvement. Users may or may not have a say in purchasing decisions, but their adoption and engagement are essential for the Product's continued success. Sometimes, the user is not the same as the end-user, e.g., in B2B2C or B2B2B. For a successful Scrum adoption, it's crucial to have regular intentional interactions between users (and possibly end-users) and the Scrum Team.

A decision-maker (called a 'chooser' by Jeff Patton) (82) is a Stakeholder with the authority to approve, select, or authorize the adoption or purchase of the Product. The decision-maker is responsible for evaluating options and making the final call, often considering the needs of both users and the broader organization. Decision-makers may or may not use the Product themselves, but their choices directly impact which Products are adopted and how value is delivered to other Stakeholders. For a successful Scrum adoption, it's often better to proceed with imperfect information, and capture emerging result feedback.

Legislators (plus, for the purpose of this document, external or internal policy makers) establish rules, policies, and boundaries for a Product's operation. They define legal, regulatory, or organizational frameworks that shape the Product's value delivery to Stakeholders and professionalism standards. They ensure the Product aligns with external or internal mandates, guiding its evolution and sustainability. For a successful Scrum adoption, it's crucial not to exaggerate or underestimate legal requirements.

Financial sponsors provide funding and resources for Product development, launch, and improvement. They assess the Product's viability, value, and feasibility, investing informed by its potential to deliver continuous value to Stakeholders. Financial sponsors influence the Product vision, strategy, and goals to ensure alignment with return on investment and long-term sustainability. For a successful Scrum adoption, it's crucial to have a flexible attitude and flexible funding as new information comes to light.

Subject matter experts contribute deep knowledge or unique skills essential to Product creation, evolution, and maintenance. Whether focused on technology, design, compliance, or a specific domain, subject matter experts support the Product's usability, feasibility, professionalism, and extendability but do not get in the way of self-managing Scrum Teams (49). They can help address satisfaction gaps and contribute to the Product's ability to adapt and identify, represent, or measure emergence (71). For a successful Scrum adoption, it's crucial to foster regular transfer of learning from subject matter experts to and across the Scrum Team.

The term satisfaction gap means the difference between what Stakeholders experience now and what they wish their experience was. In other words, it's the gap between how satisfied Stakeholders are with a Product today and how satisfied they could be.

Governance refers to structures, standards, regulations, norms, processes, and practices that consciously constrain and guide a Product's direction, decision-making, and accountability. Governance fosters transparency and guides adherence to standards of value, viability, and professionalism. It provides mechanisms for managing risks and adapting the Product to changing needs or environments, supporting its long-lived and evolutionary nature. For a successful Scrum adoption, it's crucial that governance is coherent with Scrum, e.g., a single point of contact per governance area, who has intentional interactions around quality and compliance with the Scrum Team, regular Inspection and Adaptation of the governance, and no surprises.

0.8.2 Supporter

Supporter is a specific Stakeholder type. Supporters are supporting Stakeholders and change agents. Supporters are often part of a powerful guiding coalition (83), who inspire and remove demotivating factors. Supporters support the Scrum Team to thrive and influence the organization's workflows, processes, systems, Products, services, and work environment to become coherent with a Scrum adoption and emergence (71). Supporters should participate when and where needed or as requested. Value creation often requires effective and constructive collaboration with other Stakeholders.

Depending on the size of the organization, examples of Supporters include (but are not limited to) colleagues, decision-makers, financial sponsors, governance oversight, managers, subject matter experts, marketing, HR, finance, procurement, and early adopters. Supporters who do not empower Scrum Teams to do what is recommended in this document are not really Supporters. Executives and board members have a crucial role in fostering a climate where Supporters support. Supporters should demonstrate leadership that the Scrum Team appreciates.

0.8.3 Artificial Intelligence

Artificial Intelligence (AI) is increasingly part of the work environment and may significantly expand a Scrum Team's capabilities in discovery, decision-making, Product development, and value realization.

AI may enhance Scrum through:

- Empirical Process Control (64-66): AI-driven analytics improve transparency, inspection, and adaptation.
- Cognitive Augmentation: AI allows human Scrum Team members to focus on strategic, creative, and ethical considerations.
- Continuous Value Adaptation: AI could update and reprioritize Product Backlog Items informed by live user feedback and trends.
- Systems Insight: AI identifies hidden interdependencies, improving data-informed decision-making.

The possibilities are endless. Scrum Teams could leverage AI to:

- Discover ambiguities in text and continuously inspect its own recommendations and results for bias, errors, and unintended consequences.
- Regularly validate and adapt models and applications.
- Foster transparency in Product Backlog ordering (sequencing).
- Create agents as AI team members.
- AI can be helpful to deliberately test and challenge the existing thinking.

The dangers are also endless. Maintain clear human accountability for all outcomes (guided by the accountabilities from Scrum), using AI as a powerful but supervised decision-making partner. This is known as keeping the 'human in the loop.' While AI can enhance innovation and effectiveness at the lowest costs, it does not replace human accountability. AI should support—not override—Scrum's empirical process control (64-66) and ethical (57) decision-making. The Scrum Team remains accountable for delivering valuable outcomes, assessing evidence, and upholding professionalism.

AI can be a supporting tool if used with good intent. AI tools should be evaluated like any other contributor to psychological flow (70) and learning: Do they improve feedback loops? Do they help validate assumptions faster? Do they act, and when they do, is it ethical (57) action?

Psychological flow (70) is a state of complete absorption and enjoyment in an activity, where action and awareness merge, and time seems to pass differently.

Scrum encourages the Scrum Team to experiment with AI responsibly using small, sometimes reversible trials. Adaptation and inspection apply not only to the Product but also to how AI is integrated into delivery.

The focus should remain on the human dynamics of teamwork and collaboration, with AI positioned as a potential aid.

0.8.4 Product Developer

'Product Developer' is a role and an accountability. All Product Developers together should possess all the skills needed to create Increments. The combined skill set is often referred to as cross-functional.

A Product Developer may be human or automated. Human Product Developers are *Committed* to creating, researching, inspecting, and adapting any aspect of a releasable Increment each Sprint. Their primary *Focus* is on the current Sprint. Some capacity is often invested into future-looking refinement and examining result feedback, side effects, or other learning.

Product Developers adhere to the Definition of Output Done and strive for net improvement. The Product Developers achieve the best results if they *Focus* solely on one Product. If, at a given point in time, the Product Owner or Scrum Master actively works on items in the Sprint Backlog, they perform that work as Product Developers.

The Product Developers should adopt appropriate behaviors depending on the situation; these include (but are not limited to) collaborator, creator, and a champion of technical quality, discovery, delivery, and value validation.

At least one Product Developer should be human. Multiple human Product Developers often improve cognitive diversity, helpful for addressing complexity.

Product Developers are always collectively accountable for:

- Creating an emergent plan in the Sprint Backlog for achieving the Sprint Goal;
- Instilling quality by adhering to and improving the Definition of Output Done;
- Creating at least one usable Increment every Sprint;
- Learning, often through data that is guided by the Definition of Outcome Done;
- Adapting their plan each day toward the Sprint Goal;
- Holding each other accountable as professionals; and,
- Net improvement.

Context matters, it is crucial to consider the specific circumstances. But as a rule of thumb, a Product

Developer who is neither willing nor ready nor able to be a professional should step down as a Product Developer.

0.8.5 Product Owner

Product Owner is a role and an accountability. The Product Owner must be human. To be effective, the Product Owner should be a leader for the Product. The Product Owner maximizes long-term value and needs to know where the value is and when it is needed. The Product Owner is expected to work at all levels and across all relevant business areas. The Product Owner collaborates with Stakeholders, the Scrum Master, and the Product Developers to create value.

The Product Owner uses the Product Backlog to define what gets built and in what approximate order. The Product Owner always keeps the Product Goal in mind; their primary *Focus* is to maximize long-term value at every step.

The Product Owner is not defined by analyzing and writing detailed Product Backlog Items. Every minute spent not trusting the Product Developers is a missed chance to think more strategically, work more with Stakeholders, or create more value. The Product Owner should adopt appropriate behaviors depending on the situation; these include (but are not limited to) being a visionary, customer representative, collaborator, influencer, experimenter, decision maker, and champion of Stakeholder engagement, clarity, Product quality, and value realization.

The Product Owner is always accountable for:

- Stakeholder engagement, understanding Stakeholders, their power, expectations, needs, and wants;
- Continuously sensing, listening, learning, and adapting as needed;
- Continuously balancing trade-offs, including but not limited to:
  - Quality, speed, capability, risk reduction, value, simplicity (149);
  - Stakeholders and their multiplicity of often competing expectations and limits;
  - Value, work climate, potential customers;
- Developing and explicitly communicating the Product Goal;
- Developing and effectively communicating a coherent Product narrative;
- Creating and clearly communicating Product Backlog Items;
- Ordering Product Backlog Items;
- Ensuring that the Product Backlog is transparent and understood;
- Effectively communicating outcomes supported by measures in the Definition of Outcome Done;
- Having the final say on the Definition of Outcome Done;
- Fostering the high-quality creation, discovery, delivery, and validation of value; and,
- Other Product management activities as required.

The Product Owner may do the above work or collaborate with the Scrum Team to mutually agree on responsibilities for doing the above work. Regardless, the Product Owner remains accountable.

For Product Owners to succeed, all Stakeholders and Supporters should *Respect* their decisions. These decisions are visible in the content and ordering of the Product Backlog and through the inspectable Increment at the Sprint Review. The Product Owner has delegated authority from the organization.

Product Ownership requires strong Product management skills and domain skills. A Product Owner lacking these skills may need support and guidance until they develop the necessary expertise. Context matters. But as a rule of thumb, a Product Owner who is neither willing, ready, nor able to gain Product management skills should step down as Product Owner. A domain subject matter expert is not necessarily the best choice for a Product Owner as Product management skills and leadership are needed; for example, the Product Developer accountability is often more appropriate.

If the Scrum Team inadvisably works on multiple Products, platforms, or projects, each Product, platform, or project manager should be a Stakeholder (and Supporter) of the Product Owner and collaborate to maximize long-term value. Scrum encourages the Scrum Team to stay *Focused* and *Committed*, helping it deliver valuable outcomes and avoid the pitfalls associated with operating as a 'feature-factory.'

The Product Owner is one person, not a committee or technology. Those wanting to change the Product Backlog need to convince the Product Owner. The Product Owner maximizes long-term value and often makes trade-offs in doing so.

0.8.6 Scrum Master

The Scrum Master is a role and an accountability. The Scrum Master must be human. The Scrum Master is a change agent who works at all organizational levels and across business areas. The Scrum Master leads by example and guides the effectiveness of the Product Owner, Scrum Team, Stakeholders, and Supporters in their adoption of Scrum. The Scrum Master understands complexity (30-35) and is skillful in enabling the next right thing.

The Scrum Master should adopt appropriate behaviors depending on the situation; these include (but are not limited to) being a guide, coach, mentor, teacher, observer, impediment remover, agent of change, effectiveness facilitator, and continuous improvement champion. The Scrum Master is neither a team administrator, status manager, taskmaster, rule-dictator, meeting-room booker, report dashboard administrator, chairperson, hero, coordinator, nor an in absentia Scrum Master, leaving everything to 'self-management.'

The Scrum Master is accountable for the effectiveness of the Scrum Team, Stakeholders, Supporters, and the affected people in embracing empiricism (67), self-management, and Scrum adoption as described in this document. The Scrum Master addresses whatever impedes or slows a Scrum Team's progress and cannot be resolved by the Scrum Team.

The Scrum Master supports the Scrum Team, Product Owner, and Supporters in several ways, including:

- Helping everyone understand Scrum theory and practice, educating or coaching when needed;
- Enabling the Scrum Team and Supporters to improve in a variety of ways continuously;
- Fostering timely, purposeful, and intentional interactions;
- Ensures the Scrum Team has a suitable Definition of Output Done;
- Ensuring that all Scrum events take place and are constructive, productive, and kept within the timebox;
- Causing the removal of impediments to Product-related work and to effective Scrum adoption;
- Coaching toward self-management (49) and cross-functionality;
- Helping the Scrum Team, Stakeholders, and Supporters understand their importance in supporting high-value Increments that meet the Definition of Output Done toward the Product Goal and Definition of Outcome Done;

- Improving adaptiveness (80) and optimizing the flow of value;
- Fostering evidence-informed confidence but being compassionate and timely to avoid overconfidence;
- Fostering change agency and general agency by the Scrum Team and Supporters;
- Encouraging helpful behaviors within the Scrum Team that are aligned with the Scrum Values to foster trust, collaboration, and high performance; and,
- Fostering the Scrum Team to deliver valuable work, get feedback, and do rework as needed, quickly and often.

The Scrum Master supports the Scrum Team in several ways, including:

- Supporting the Scrum Team in its formation, upskilling, and continuous improvement;
- Helping the Scrum Team understand the need for clear and concise Product Backlog Items that deliver value; and,
- Being vigilant that the entire Scrum Team is collaborating purposefully and intentionally with each other and Stakeholders, honoring the Definition of Output Done, and focused on creating high-value Increments according to the Definition of Outcome Done.

The Scrum Master supports the Product Owner in several ways, including:

- Helping find techniques for effective Product Goal definition and Product Backlog management;
- Helping establish emergent Product planning for a complex (30-35) environment;
- Helping the Product Owner to express outcomes as measures through the Definition of Outcome Done;
- Helping the Product Owner understand the need for clear and concise Product Backlog Items that deliver value; and,
- Helping the Product Owner to *Focus* on value realization.

The Scrum Master supports the Stakeholders in several ways, including:

- When more than expertise is needed, helping affected people and Stakeholders understand and adopt:
  - An empirical approach for complex (30-35) work where cause and effect are only coherent in retrospect,
  - Going beyond empirical process control, e.g., running multiple parallel safe-to-fail experiments, seeking fresh thinking, exaptation, or testing educated hunches. Exaptation means taking something that was made or used for one purpose and using it for a different purpose, especially when facing new or unclear situations.
- Fostering actions that support the mantra ‘Stop putting items in progress; start finishing items;’
- Facilitating Stakeholder collaboration as requested or needed;
- Helping Stakeholders understand the need for clear and concise Product Backlog Items that deliver value; and,
- Helping the Stakeholders to *Focus* primarily on value realization.

The Scrum Master works with the Supporters in several ways, including:

- Leading, training, and coaching the Supporters in the Scrum adoption;
- Clarifying what is getting in the way of an effective Scrum adoption;
- Facilitating disciplined emergent change in a direction to support the Scrum adoption; and,
- Fostering organizational changes toward ease of delivery vs ease of management.

The Scrum Master works with the organization in several ways, including:

- Leading, training, and coaching the organization in its Scrum adoption(s);
- Planning and advising Scrum adoptions within the organization;
- Working with related departments in how they could support the Scrum adoption; and,
- Removing barriers to the Scrum adoptions.

Scrum Masters can team up with other Scrum Masters or Supporters to support the whole organization; they can also collaborate with other change agents or leaders when needed. The Scrum Master, as a change agent, is accountable for the quality of the Scrum adoption and should collaborate with other change agents to improve it.

The Scrum Master is one person, not a committee or technology, and serves the Product Owner, Scrum Team, Stakeholders, and the larger organization. Being a change agent and leader, the Scrum Master should generally invite people to participate in the change. It is helpful if the Scrum Master has an understanding of the flow of value (68,69), lean (63), complexity theory (30-35), and other supporting and complementary theory in this document, as well as assisting people with the *how*. It is also helpful if the Scrum Master is unrelenting and has an insatiable appetite for learning and change.

Being a Scrum Master is a calling where helping others succeed is reward enough. A Scrum Master doesn't seek the spotlight. Like any good leader, they give credit to others and take responsibility when things go wrong. Staying in the role for a longer time helps guide the Scrum Team toward its full potential, but only if the Product Developers collectively develop self-management. Parent-style Scrum Master behavior does not foster a self-managing Scrum Team. Context matters. But as a rule of thumb, a Scrum Master who is neither willing, ready, nor able to be an agent of change should step down as a Scrum Master.

0.9 The Scrum Artifacts in the Expansion Pack

Scrum's artifacts provide Transparency about what the Scrum Team and Stakeholders believe will deliver value. Thus, everyone can have the same basis for Inspection and Adaptation.

Each artifact contains a commitment:

- For the Product serving the Stakeholders, it is the Definition of Outcome Done.
- For the Increment that is a candidate update for the Product, it is the Definition of Output Done.
- For the Product Backlog, it is the Product Goal.
- For the Sprint Backlog, it is the Sprint Goal.

Upon release of the Increment (output), the Product is what creates value (outcomes). Value is the measurable or observable fulfillment or creation of expectations, needs, or wants from the Stakeholders' perspective.

These commitments reinforce the pillars of Transparency, Inspection, and Adaptation, enabling empirical process control (64-66). The Product Goal is fixed for as long as no contrary evidence or observations emerge in the observed Product's Definition of Outcome Done. The Definition of Output Done is not weakened during the Sprint. So what could be diminished or changed instead? It could be the Acceptance Criteria for a specific Product Backlog Item, the implementation or fidelity of a specific feature, or even alternative Product Backlog Items for achieving the Sprint Goal, etc.

If the Product Goal shifts often, it could indicate that something is off, perhaps due to a lack of *Focus* on what matters. *Focus* is about being professional and deciding what to work on but also what not to work on.

0.9.1 Product

The Product is an artifact. A Product can be a holistic experience or a platform. It can also be a service, physical, digital, or hybrid, delivering continuous value to Stakeholders (including but not limited to users).

An experience is a specific solution designed to meet the needs of Stakeholders, including the user, ideally external to the organization. It provides a direct interaction that delivers value. It is typically focused on solving a particular problem or opportunity, or a set of them for Stakeholders, including but not limited to customers, decision-makers, and users.

A platform is an architectural device, foundational infrastructure, or set of tools that enables developers to build Products in order to provide an experience. Platforms provide a base for multiple Products to be developed upon, focusing on scalability, reliability, and flexibility for engineers rather than direct user interaction.

The Scrum Team and Stakeholders need to have a clear understanding at all times of what the Product is, who the customers, users, or decision-makers are, and the type of Product it is —like one for end-users, employees, or Scrum Teams—has different Stakeholders and ways it creates value. A Product is evolutionary and often long-lived. The Product needs a single Product Backlog to increase Transparency and maximize value.

Context matters. But as a rule of thumb, for a Product to create and maintain traction, it helps if the Product:

- Sufficiently addresses satisfaction gaps;
- Is valuable, desirable, viable, usable, feasible, safe, and secure;
- Has professionalism built-in;
- Has a compelling, clear, and outcome-metric-oriented Product Vision, Product strategy, and Product Goal, often including intent, rationale, and anti-goals;
- Adapts and improves to identify, represent, or measure emergence (71); and,
- Is extendable and maintainable.

The Product is the manifestation of *why* we do *what* we do.

Commitment: Definition of Outcome Done The Definition of Outcome Done is a commitment. It describes the observable evidence measures (quantitative or qualitative) required to demonstrate realized benefits, often referred to as value validation. It could be for the overall Product or a specific goal. It's

often best to define the measures for value validation before realization starts, as this avoids biases and mistaken interpretations.

Outcomes and related interpretations inform future adaptations, ideally confirming the intended Stakeholder impact(including but not limited to business or user impact)—measuring whether the output fulfills the anticipated outcome(s) and delivers real value. It could be for a specific goal, such as a larger feature or several features, and be validated through Product telemetry (the Product can measure its own usage). Alternatively, it could be for the overall Product, where it is often about the strategic impact and the validation of the efficacy of the implemented strategic deployment (120-124). Or a combination of both.

Favor direct evidence over circumstantial evidence. For example:

- Customer outcomes could Focus on delivering measurable value to customers, such as increased customer satisfaction, customer long-term cost reduction, or the number of customer jobs addressed.
- User outcomes could address specific changes in user behavior that solve problems and improve experiences, like completing tasks more efficiently or engaging with new features.
- Product Stakeholder outcomes could connect these behavioral changes to Product performance metrics, e.g., trends in Product customer, decision-maker/user metrics, Product time to release, time to learn, time to pivot, etc.
- Business Stakeholder outcomes, e.g., compliance, business long-term cost reduction, business results, trends in market share, customer satisfaction across all Products, organizational time to release, time to learn, time to pivot, etc.
- Scrum Team outcomes such as improved technical capability (psychological flow (70), frequency of release, tooling, skills, technical debt, UX or CX debt, capacity), climate/culture for net improvement and innovation.

User eXperience (UX) or Customer eXperience (CX) debt is the sum of design and implementation choices—intentional or not—that make a Product or service less usable, enjoyable, or effective for users or customers. Recognizing, tracking, and addressing this debt is essential for delivering Products that truly meet user needs and expectations.

Measures over time make Product, market, and Stakeholder trends (including but not limited to customer or user) transparent; these can be reviewed at any time during the Sprint, including the Sprint Review.

0.9.2 Increment

The Increment is an artifact. It is the integration of the work completed to the standard of the Definition of Output Done. The Increment is an output and a Product candidate.

Multiple Increments may be created within a Sprint through the completion of Product Backlog Items. Each Increment is thoroughly verified, usable, and integrated with all previous Increments. The resulting aggregated Increment is inspected as soon as possible, at the latest at the Sprint Review. The Increment must be usable and useful to enable result feedback. An Increment is central to Scrum as it enables ongoing value validation.

An Increment-candidate does not qualify as an Increment until it meets the quality standard of the Definition of Output Done. Only an Increment can be released. An Increment should be a concrete stepping stone toward the Product Goal. Increments may be delivered to Stakeholders or released prior to the Sprint Review. *The best value validation is attained through result feedback.*

Commitment: Definition of Output Done The Definition of Output Done is a commitment. It formally describes the quality measures that express due diligence for the Increment so that it could be delivered to Stakeholders.

The Definition of Output Done typically includes (but is not limited to) both technical standards and Product qualities. The Scrum Team creates it if not provided by the organization as a minimum. If there are multiple Scrum Teams on the same Product, they share the same Definition of Output Done as the common foundation but may improve upon it.

The Scrum Team is duty-bound to conform to the Definition of Output Done and continuously improve it. The Increment is cumulative. The Definition of Output Done is for the good of the Product and its Stakeholders. The Definition of Output Done is the overall quality standard for the whole Increment, not the specific standard for each item (e.g., Acceptance Criteria).

A released Increment enables result feedback for Definition of Outcome Done value validation.

0.9.3 Product Backlog

The Product Backlog is an artifact. It is the emergent, ordered (sequenced) list of Product Backlog Items needed to attain the Product Goal. The Product Backlog provides Transparency (work clarity) and is the single source of work for the Scrum Team in order to achieve the Product Goal. The Product Owner, always keeping value in mind, guides the ordering of the Product Backlog Items in the Product Backlog. A smaller Product Backlog often provides more Transparency.

Product Backlog Item A Product Backlog Item is a potentially valuable item in the Product Backlog. It is not necessarily in any specific format. It is intended to deal with a problem or opportunity. It can include Acceptance Criteria that can tell when work is completed, in addition to the Definition of Output Done. One might deliver exactly what was requested but still not deliver sufficient outcomes. So, a Product Backlog Item can also include clearly defined Outcome Criteria that can tell when sufficient value is delivered, in addition to what is already in the Definition of Outcome Done.

A Product Backlog Item is a single piece of work that discovers or delivers value. A Product Backlog Item can evolve anytime, even while Product Developers work on it. During refinement, it is broken down into smaller, more understandable (mostly to the Scrum Team) Product Backlog Items that could deliver value. Occasionally, an item in the Product Backlog might not be related to the Product Goal; if this happens often, it's worth examining if the *Focus* level might not be where it needs to be. The Scrum Team and Stakeholders should *Focus* on outcomes over outputs, keep the trade-off balance right, and not let the Scrum Team become a 'feature-factory' or 'discovery-factory.'

Acceptance Criteria Acceptance Criteria, if they exist, describe when an output for a specific Product Backlog Item is complete, in addition to the Definition of Output Done. Acceptance Criteria in refined items should provide unambiguous clarity on *what* is requested. Acceptance Criteria include criteria specific to a Product Backlog Item not already addressed in the Definition of Output Done; they can be functional or non-functional. Acceptance Criteria can evolve anytime, even while Product Developers work on them.

Outcome Criteria Outcome criteria, if they exist, describe the intention of the Product Backlog Item; it is the *why* behind the *what*. The fulfillment of Outcome Criteria often complements the Definition of

Outcome Done for the Product. They can include criteria specific to a Product Backlog Item not already addressed in the Definition of Outcome Done. If questions arise, the Outcome Criteria provide direction; they can be in the form of a narrative or measures, ideally, the latter. Outcome Criteria can evolve anytime, even while Product Developers work on them.

Refinement Refinement is an activity. It may be formal (an additional event) or informal. Refinement is an ongoing emergent process that fosters clarity and reduces risk; it builds enough understanding and confidence that the selected or upcoming Product Backlog Items are ready (can be completed in accordance with the Definition of Output Done within a small number of days, or shorter). Various types of dependencies are considered.

Refinement involves breaking down Product Backlog Items into smaller, more understandable (mostly to the Scrum Team) Product Backlog Items. It can add more details such as description, Acceptance Criteria, Outcome Criteria, order, and size. Attributes vary but should be meaningful to the Scrum Team. Refinement can involve research, including but not limited to, problem or opportunity validation, user or customer experience, solution validation. The Product Developers, and nobody else, are responsible for sizing the Product Backlog Items. The Product Owner may influence the Product Developers by helping them understand and select potential trade-offs.

Participants often include Stakeholders and members of the Scrum Team; it is not uncommon for Product Developers to work directly with Stakeholders. Refinement is often supported or facilitated by the Product Owner. The Product Owner can *Focus* more on Product ownership if the Product Developers have a broad understanding of the Product. Generally speaking, it is a forward-looking activity that offers clarity, direction, and potential *Focus* for upcoming Sprints.

Commitment: Product Goal The Product Goal is a commitment. It is represented through the Product Backlog, which is owned by the Product Owner. It is the current single, more strategic, ambitious objective (the *why*). It gives direction for the Product and enables *Focus* for the Product Developers working on the Product. It improves Transparency by providing a clear, valuable direction for the Product Developers to work toward, using a more tactical Sprint Goal (the *why* for the Sprint).

A Product Goal is the medium-term objective for the Scrum Team and the Stakeholders (and Supporters). The Scrum Team should fulfill (or abandon) one Product Goal before taking on the next.

A Product Goal is usually an as-yet-unproven assertion about value. It can be expressed as one of many things, including a set of hypotheses about closing or lessening satisfaction gaps. It gets the balance right by focusing on a subset of the multiplicity of Stakeholders (including but not limited to customers or users) expectations and limits. Through Inspection and Adaptation, it's essential to embrace uncertainty (72), result feedback, side effects, and other learnings.

What about a Product Vision? {#what-about-a-product-vision?} Many organizations work with a Product Vision, which helps visualize a potential future. The Scrum Team can use a Vision as input for considering a Product Goal. A Product Vision is a meaningful, long-term set of valuable desired outcomes. The medium-term Product Goal is often a stepping stone toward a long-term Product Vision.

As the Scrum Team and Stakeholders inspect and adapt toward the Product Goal, they need to be *open* to the idea that the Product Vision or Product Goal might also need to adapt. Often, several Product Goals are sequentially achieved while working toward a vision.

The key thing to note is that a Product Vision is often a work of fiction; none of it may be true. Forming hypotheses and running experiments in a direction is essential, and is where Scrum can add the most value.

A Product Vision is often inspiring but can be overwhelming. The Product Goal reduces overwhelm by acting as a more tangible vertical slice of a Product Vision or as an enabler for a Product Vision.

0.9.4 Sprint Backlog

The Sprint Backlog is an artifact. It is composed of the Sprint Goal (the *why* for the Sprint), the set of Product Backlog Items selected (the *what*, also known as the forecast) for the Sprint, and often has an actionable plan for delivering the Increment (the *how*). It provides Transparency (work clarity) throughout the Sprint.

The Sprint Backlog is a plan by and for the Product Developers. It is the Product Developers' viewpoint of the understood work to achieve the Sprint Goal (the why for the Sprint). Suppose a suboptimal scenario where most items in the Sprint Backlog are continually unrelated to the Product Goal. In that case, the *Focus* and *Commitment* Scrum Values are not being upheld.

Within the context of the Sprint Goal, the Product Developers update their plan, even the forecast, throughout the Sprint as more is learned. The Sprint Backlog should have enough work to get started, e.g., start with one or two Product Backlog Items toward the Sprint Goal. The Product Developers inspect their progress toward the Sprint Goal in the Daily Scrum or more often. Product Developers learn to adapt and respond to uncertainty (72).

Commitment: Sprint Goal The Sprint Goal is a commitment created and owned by the Scrum Team. The Sprint Goal is the single unifying objective of the Sprint (the *why*) for the Product Developers, created in Sprint Planning. Delivery of the Sprint Goal is a commitment by the Product Developers. The Sprint Backlog (including the *why*, the *what,* and, often, the *how*) provides *Focus* and flexibility regarding the evolving work, thus improving Transparency.

The Sprint Goal encourages the Scrum Team to work together rather than on separate initiatives. If the work turns out to be different from what the Product Developers expected, the Product Developers collaborate with the Product Owner to negotiate possibilities within the Sprint without affecting the Sprint Goal. No one tells the Product Developers how to size or do their work.

0.10 The Scrum Events in the Expansion Pack

Scrum combines four timeboxed events for Inspection and Adaptation within a containing fifth event of determinate length, the Sprint. These events support the Scrum pillars of Transparency, Inspection, and Adaptation. Releases enable value, ideally, continuously. Infrequent releases lead to delayed result feedback.

A timebox is a stipulated maximum amount of elapsed time from beginning to end for a defined event, not to be confused with an expectation to use that full amount of time. The purpose of a timebox in Scrum is to foster the selection of essential work, creating *Focus* to achieve desired results quickly. In Scrum, for a given Scrum Team, the sprint length is consistent, so it is not a timebox.

Events create cadence and minimize the need for other meetings not part of Scrum. Ideally, each event is held at the same time and place to reduce complexity (30-35) and foster the formation of habits.

Skilled facilitation improves effectiveness. Ineffective events risk losing emphasis on the Sprint Goal, Product Goal, Transparency, Inspection, Adaptation, and Scrum Values.

Each event has its own purpose and should include deep, meaningful work. Together, the Scrum events provide a scaffold of Transparency to inspect and adapt, pause, and reflect. The Scrum events support structured thinking and working, effectiveness, and a balanced workload.

Communication is key to ensuring the Scrum Team and Supporters *Focus* on the right thing. Apart from the Sprint, events may consume less time as long as coherence is not lost.

0.10.1 The Sprint

The Sprint is an event where ideas are turned into value. The Sprint is the container event. It is an iteration of a determinate time in which work is carried out. It provides *Focus* and stability. A Sprint is no longer than four weeks. A new Sprint starts immediately after the conclusion of the previous Sprint. All the work necessary to achieve the Product Goal happens within Sprints.

Sprints are the heartbeat of Scrum, where the Scrum Team turns ideas into usable, useful, and potentially valuable Increments. The Increment is released as soon as practically possible, considering the need for early result feedback. A lack of release to some subset of Stakeholders (including but not limited to real customers, decision-makers, and users) can lead to a lack of timely result feedback. Multiple Increments may be created in a Sprint; the Scrum Team should strive to validate value through early and frequent releases, where applicable.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- The Increment(s) should not decrease in quality;
- The Product Backlog is refined as needed; and,
- As more is learned, current work may be clarified and renegotiated with the Product Owner without affecting the Sprint Goal.

Sprints enable outcomes by ensuring Inspection and Adaptation of progress toward a Sprint Goal at least every four weeks. When a Sprint is too long, the Sprint Goal may become invalid, increasing complexity (30-35) and risk. Shorter Sprints often generate more learning cycles; they could also limit risk.

Shorter Sprints usually require improved capabilities (e.g., refining, vertical slicing, technical domain, business domain). Context matters and the Scrum Team strives to strike the right balance.

Various complementary practices exist to assess or forecast progress, like burn-downs, burn-ups, flow analytics, Monte Carlo probabilistic forecasts, large effort estimation, fuzzy sets (110), etc. While useful, these do not replace the importance of empiricism (67). In complex (30-35) environments, what has already happened may be used for forward-looking decision-making, but what will happen is unknown.

You could think about a Sprint as a mini project with a clear outcome, determinate length and understood costs. However, the various work activities happen in parallel and not in a sequential defined linear way.

A Sprint could be canceled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint. Shorter Sprints lower the likelihood of a cancellation.

0.10.2 Sprint Planning

Sprint Planning is an event. This first event of the Sprint is where the Scrum Team gives *Focus* and creates commitment.

During Sprint Planning, the more strategic Product Goal (the *why* for the Product Backlog) is considered and provides direction. In doing so, the Product Developers create the Sprint Backlog, which consists of the short-term, more tactical Sprint Goal (the *why* for the Sprint), the initially identified work, and the plan to deliver.

Sprint Planning addresses the following topics:

The *Why* for the Sprint The Product Owner proposes ideas for how the Product could increase its value and utility in the current Sprint. The Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to Stakeholders toward the Product Goal. The Sprint Goal must be finalized by the end of Sprint Planning.

The *What* toward the *Why* Through collaboration with the Product Owner, the Product Developers select items from the Product Backlog to include in the current Sprint. The Scrum Team may refine these items, which increases understanding and confidence. Selected items should be achievable according to the standard of the Definition of Output Done, alongside other items.

Selecting how much can be completed within a Sprint may be challenging. However, the more the Product Developers know about their past performance, vertical slicing, upcoming capacity, and the Definition of Output Done, the more confident they will be in their ability to forecast Sprint outcomes.

Successful Scrum Teams do not overload themselves. In fact, they plan to finish work early, sometimes using a buffer for unexpected events (85). This helps the Scrum Team to stay focused, improve quality, and satisfy Stakeholders by delivering value sooner. Chronic overload or sudden shifts can cause excessive negative stress, which Jeff Sutherland calls 'Bayesian surprise.' They can disrupt the Scrum Team's psychological flow (70) and performance. Clear communication, professional handling of emergence (71), and small, regular changes help prevent this, so Scrum Teams should aim for early delivery.

The *How* for the *What* How the work is done is at the sole discretion of the Product Developers. No one else tells the Product Developers how to do their work. The Product Developers select their own work; no one else assigns or pushes Product Backlog Items to the Product Developers, not even the Product Owner.

Sprint Planning is timeboxed to a maximum of eight hours for a four-week Sprint. The event is usually shorter for shorter Sprints. Context matters. But as a rule of thumb, do enough planning to get started with the work, e.g., plan a few Product Backlog Items toward the Sprint Goal.

0.10.3 Daily Scrum

The Daily Scrum is an event. At the Daily Scrum, the Product Developers collaborate on progress toward the Sprint Goal and update the actionable plan, the Sprint Backlog, until the next Daily Scrum. In the event the Sprint Goal has already been achieved, the Product Developers collaborate on meaningful progress toward the Product Goal.

The Daily Scrum provides *Focus*, cohesion, and urgency and fosters self-management (49). Usually, only the Product Developers participate. To simplify, it often uses the same meeting cadence, place, and

time.

The Product Developers can select whatever structure and techniques they want. Daily Scrums improve communication towards attaining the Sprint Goal, identify and address risks and impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

The Daily Scrum is not the only time the Product Developers adjust their plan for the Sprint within the context of the Sprint Goal or Product Goal. Product Developers often meet throughout the day for more detailed discussions.

To enable the flow of value (68,69) and enable potential outcomes sooner, the Product Developers should focus on one item or a few items at a time and meet the Definition of Output Done, before starting to work on other items. The Product Developers can achieve this by focusing, having fewer items in progress, and proactively finishing work over starting new work. The Product Developers monitor idle work, not idle people.

The Daily Scrum is timeboxed to a maximum of fifteen minutes per day; it may be shorter.

0.10.4 Sprint Review

The Sprint Review is an event. It is an interactive, collaborative working session. Often, the Scrum Team shares the current Product Goal and presents the Definition of Output Done and the Definition of Outcome Done to the Stakeholders. The Scrum Team shares the results of their work, what trade-offs were made, and how much progress was made toward the Product Goal (the why behind the work). If available, current and up-to-date measures of progress toward the Definition of Outcome Done are shared and considered.

The Sprint Review inspects many things related to the Product, such as the Product Goal, Product Backlog, the Sprint Goal, the learnings, the Increment, Stakeholder expectations and limits, result feedback, side effects, progress with the Product, the market, as well as forward-looking, e.g., what new ideas and opportunities have emerged, potential next steps.

Informed by what is learned:

- Participants sense, listen, learn, and collaborate on what to potentially do next;
- The Product Backlog (the what)is adapted and possibly the Product Goal, ideally supported by evidence or observations and guided by the Product Goal or optional Product Vision; and,
- Participants adapt the Product's Definition of Outcome Done for future Sprints.

It's always important to consider Stakeholders and what they value, including inanimate, non-human Stakeholders such as the law.

Incomplete Product Backlog Items return to the Product Backlog for future consideration and are not presented; sometimes, they are moved into the next Sprint.

The Sprint Review is the second-to-last event of the Sprint and is timeboxed to a maximum of four hours for a four-week Sprint. For shorter Sprints, the event is usually shorter.

0.10.5 Sprint Retrospective

The Sprint Retrospective is an event. At this event, the Scrum Team agrees on how to improve. Bad assumptions are also explored, i.e., assumptions that led the Scrum Team in the wrong direction. Good things like particular technologies, processes, patterns, etc., might also be pointed out or reinforced.

Inspected elements often vary with the domain of work. Reflection is more effective in a psychologically safe environment.

The Sprint Retrospective focuses on the most helpful changes to improve, such as:

- The Increment
- Outcomes
- Professionalism, e.g., skills, technical practices, tooling, ability to innovate;
- Flow of validated value (68,69), e.g., end-to-end flow metrics, time-to-market;
- Effectiveness (the *how*), e.g., technology, processes, dependencies;
- Interactions and Scrum Team dynamics, e.g., collaboration, working arrangements;
- Information radiators, e.g., product wall, metrics;
- The Definition of Output Done for future Sprints;
- Further adaptations to the Definition of Outcome Done for future Sprints;
- How to automatically attain the measures regarding the Definition of Outcome Done;
- And more.

The most impactful improvements should be addressed as soon as possible. The Scrum Team should not just talk about improvement; Scrum depends on meaningful, continuous improvement follow-through. Some improvement actions rely on the assistance of Supporters, but that does not mean the Scrum Team should not strive for net improvement regardless (such as continual marginal gains).

The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a four-week Sprint. For shorter Sprints, the event is usually shorter.

0.11 Multi-Scrum-Team Products

If a Scrum Team becomes too large, it should consider reorganizing into multiple cohesive Scrum Teams, each focused on the same Product. Multiple Scrum Teams on the same Product should share the same Product Goal, Product Backlog, Product Owner, baseline Definition of Outcome Done, and baseline Definition of Output Done.

Be careful with blind assumptions that more value is produced from more Scrum Teams. Scale only when the benefits clearly outweigh the additional overhead. Before you scale, the single Scrum Team setup has to be able to reliably produce an Increment every Sprint. But if you must scale, use an approach that is coherent with this document. Often fewer teams result in more outcomes.

In a multi-Scrum-Team context, Scrum Teams may reduce inter-Scrum-Team dependencies by becoming more cross-functional through collaboration, cross-pollination, transfer of learning, and intentional interactions. The specific skills needed are often broad and will vary with the domain of work. In a multi-Scrum-Team setting, purposeful and intentional interactions and professionalism (including continuous integration) become even more important.

In a one Product Owner and one Scrum Team setup, the Product Owner could be a Product manager, marketing director, technology director, etc. In a multi-Scrum-Team setup for a Product, the ideal is still only one Product Owner, who should be a leader for the Product. To allow the Product Owner to handle multiple Scrum Teams on the same Product, the Product Owner often becomes more strategic and delegates problems to solve and opportunities to the Product Developers, including, for example, aspects of Product design or Product management.

The Product Backlog is a tool for increasing transparency.

Generally, the fewer Product Backlogs per Product, implicit (like a filter of a Product Backlog) or explicit:

- The fewer the silos in the Product and the greater the transparency across the entire Product;
- The more transparent the overall progress tracking across the entire Product;
- The better the big-picture value clarity across the entire Product;
- The more likely a Scrum Team would know it's working on low-value items from a Product perspective;
- The more likely one is to observe improvement in the attainment of value; and,
- The more strategic the Product Owner becomes by delegating cross-Product work to the Product Developers.

Fewer Product Backlogs per Product are better for adaptiveness (80), but without empowered ownership, a coherent span of control, or direct contact with relevant Stakeholders, gaps will arise. Scrum fosters a climate for happenstance and multi-learning as various people and Scrum Teams collaborate, discoveries and insights can be shared and leveraged. This is unlikely to happen in an environment where each component has a Product Backlog in isolation.

Happenstance in the context of 'The New New Product Development Game' (29) means that sometimes useful ideas or solutions come about by chance, not by careful planning. When Scrum Teams work closely together and share information, they might discover new approaches or answers simply because they are open to unexpected events or accidental findings.

Multi-learning means that team members learn in many different ways at the same time. They pick up new skills and knowledge not only in their own area but also in other areas, and they learn as individuals, as a group, and as part of the whole company. This helps the team become more flexible and able to solve a wide range of problems quickly, because everyone is learning from each other and from their experiences as they work together.

Finding the right balance is a dilemma. There are always trade-offs to consider. However, a good heuristic is: The fewer Product Backlogs, implicit or explicit, the better, enabled through multi-learning and the organizational transfer of learning across Scrum Teams, departments, and Products.

Organizational transfer of learning, as described in 'The New New Product Development Game' (29), is the process by which knowledge and insights gained in one new Product development area are regularly shared and applied to subsequent areas or other divisions within the organization.

Organizations are often designed for ease of management over ease of outcomes. Ask yourself how many Scrum Teams a problem or opportunity touches to deliver value; generally, the lower that number, the better.

Free teams from command and control. Err on the side of aligned autonomy. Foster purposeful, intentional interactions within and between self-managing Scrum Teams (49). Foster a work climate with minimal but sufficient management processes, scaffolding, and boundaries. Balance and nurture Stakeholder expectations and limits. Build change agency and continual improvement in a direction, not just delivery, into a cadence.

When in doubt, study the New New Product Development Game (29), embrace the good of what's new in the present, but abandon any notions of an industrial complex (30-35) where only the brave people have the agency to do anything.

0.12 End Note

Jeff Sutherland's 1993 Scrum adoption at Easel was inspired by Christopher Langton's papers (36,37) on Complex Adaptive Systems (CAS) (74-77) theory from Los Alamos Labs, which shows that systems evolve quicker at the edge of chaos.

Scrum is described in the 2020 Scrum Guide (40). Tobias Mayer's A Simple Guide to Scrum (58) is a shortened, edited version of the official Scrum Guide by Ken Schwaber and Jeff Sutherland. The Scrum Hexis (52) elaborates on the 2020 Scrum Guide from a 2025 perspective, but the 2020 Scrum Guide is still the essential reference for Scrum.

Scrum is like a mirror. If the image in the mirror is not as expected, should the mirror be hidden?

Attain at least one Increment each Sprint as a habit before you adapt Scrum. Every part of Scrum has a purpose; understanding the why for each part is essential. Consider the context. The short-term is about delivery. The long-term is about successful emergent change in a direction and the sustainable delivery of value. Successful Scrum adoption depends on getting the balance right between the short and long terms.

Be careful about copying approaches from other organizations without also fostering their culture. Emergent change in the direction of travel is the change. The change includes (but is not limited to) leadership, workflows, processes, and systems, including HR, Finance, Procurement, and more. Scrum is part of a never-ending expedition of continual improvement and evolution in a direction of travel rather than a destination.

0.12.1 Acknowledgments

Scrum was inspired by Lean (63), the Toyota Production System (59-60), the Harvard Business Review article 'The New New Product Development Game' by Hirotaka Takeuchi and Ikujiro Nonaka (29), and Empiricism at Dupont (61).

Scrum was developed in the early 1990s. Ken Schwaber and Jeff Sutherland first co-presented Scrum at the OOPSLA Conference in 1995 (62). The first version of the Scrum Guide (40) appeared in 2009. Scrum is evolving.

We also thank reviewers who provided feedback to earlier drafts, including but not limited to, Daryn Basson, Alex Benes, Kurt Bittner, Deb Bhattacharya, Magdalena Firlit, Nichervan Fazel, Peter Fischbach, Michael Forni, Tom Gilb, Martin Hinshelwood, Jesse Houwing, Michael Huynh, Matthew Ijogi, Marc Kaufmann, Christian Neverdal, Stas Pavlov, Ian Sharp, Alisa Stolze, Mark Summers, and Nader Talai.

1

1.1 Scrum *Expanded* on One Page

Scrum is described in the 2020 Scrum Guide (40). Scrum is a lightweight framework for addressing complex (30-35) work, particularly in Product discovery, development, delivery, and value realization. Scrum is based on empirical process control (decisions informed by evidence) and lean thinking (reducing waste and focusing on the flow of value) (63). Scrum is purposefully incomplete, guiding interactions rather than prescribing detailed recipes.

Why Use Scrum?

Scrum enables Scrum Teams to identify, represent, or measure emergence (71), embrace uncertainty, respond to change, deliver and validate value frequently, and continuously improve. Scrum fosters collaboration, accountability, and evidence-informed decision-making, fostering the best possible outcomes in a rapidly changing environment. Self-managing Scrum Teams, organized around value, are crucial for creative problem-solving and opportunity capture; non-self-managing Scrum Teams hinder the ability to deal with complexity (30-35). Self-managing Scrum Teams are not to be confused with individual self-management.

Elements of Scrum

1. Scrum Theory: Built on three pillars:

- Transparency – Making work and value visible for Inspection.
- Inspection – Regularly assessing progress and outcomes for Adaptation.
- Adaptation – Adjusting plans informed by insights and feedback.

2. Scrum Values:

- *Focus*, *Openness*, *Courage*, *Commitment*, and *Respect* enable effective teamwork; they support trust.

3. *Roles* / Accountabilities:

- Scrum Team – A small, self-managing, cross-functional, cognitively diverse team consisting of:
  - Product Owner – Maximizes long-term value, engages Stakeholders, and manages the Product Backlog.
  - Scrum Master – Guides the Scrum adoption, removes impediments, and fosters continuous improvement.
  - Product Developers – Deliver Increments every Sprint through their cross-functional capabilities.
- *Stakeholder - an entity, individual, or group interested in, affected by, or impacting inputs, activities, and outcomes with a direct or indirect interest inside or outside the organization, its Products, or services.*
  - *Supporter, a Stakeholder type – Fosters the climate and environment and participates as requested.*
  - *AI – As a tool or also a possible Product Developer, but not to be entirely trusted yet.*

4. Scrum Events & Activities

- Scrum operates in Sprints (iterations of determinate length *up to four weeks*) with four time-boxed events:
- Sprint Planning – Define the Sprint Goal and plan the work.
- Daily Scrum – Product Developers align daily on progress toward the Sprint Goal or Product Goal.
- Sprint Review – Inspect the Increment, value, and marketplace, and adapt the Product Backlog.
- Sprint Retrospective – Reflect and improve the Scrum Team.
- Refinement – Clarify upcoming or selected work, formally (*as an optional event*) or informally

5. Scrum Artifacts & Commitments

- *Product & Definition of Outcome Done – Product and valuable outcomes that provide evidence of realized benefits.*
- Increment & Definition of Output Done – A potentially valuable, releasable candidate update for the Product.
- Product Backlog & Product Goal – the ordered (sequenced) list of work to achieve a medium-term, more strategic objective.
- Sprint Backlog & Sprint Goal – Selected Product Backlog Items and a plan for the Sprint, short-term objective.

2

2.1 Expansion Log

Additions

- AI section
- Self-managing Scrum Team, Cadence, Professionalism sections
- Emergence section, open to the idea that risk or variances from expectations don't necessarily go down over time
- Complexity (30-35)– The case for Scrum section
- Leadership and Systems thinking sections
- Product thinking and Discovery sections
- First principles, People, and Change sections
- Multi-Scrum-Team Products section
- Stakeholder role (including customers, decision-makers, and users), Supporter as a Stakeholder type
- Refinement, Product Backlog Item sections
- Optional: Product Vision, Acceptance Criteria, Outcome Criteria
- Definition of Outcome Done, extra emphasis on adaptation informed by outcome evidence
- Stakeholder, value, result feedback, release, outcomes, risk, impediment, and leader are defined
- Flow analytics, Monte Carlo probabilistic forecasts, large-level estimation, fuzzy sets (all optional)
- Scrum Expanded on one page
- A need to make workflows, designs, processes, systems, and the work environment coherent with emergence
- 'Product Ownership requires strong Product management skills and domain skills…A Product Owner who is not willing, ready, or able to gain Product management skills should step down as Product Owner.'
- A Product Developer who is neither willing nor ready nor able to be a professional should step down.
- A Scrum Master who is neither willing, ready, nor able to be an agent of change should step down.
- Appendices: Cynefin® Kind of Explanation - unofficial and unauthorized, Emergent Strategy, Adaptive (80) Enterprise, Adaptive Executive or Board Member

Suggestions

- Clarification and modification of responsibilities while ‘embracing fuzziness’ (73)
- From Scrum is immutable or simple to Scrum is evolving, in some cases, softened wording from ‘must’ to ‘should’
- Product Owner accountability to Product Owner role with accountability; maximizing long-term value
- Developers accountability to Product Developer role with accountability
- Scrum Master accountability to Scrum Master role with accountability; Scrum Master is one person, not AI
- Product Developers may be human or AI, or helped by AI, at least one human; more human Product Developers are better for cognitive diversity and addressing complexity
- Scrum Team commits to the Sprint Goal, not the former Developers; important that Product Owner is focused
- Sprint Backlog toward Sprint Goal or Product Goal, not just Sprint Goal
- Product definition, mention of Product strategy, roadmaps, Product models, scaling, goal-oriented approaches
- Emphasis on learning, result-feedback, side effects, outcomes over outputs
- To preserve the flow of value, incomplete Product Backlog items don’t have to return to the Product Backlog
- Definition of Done renamed to Definition of Output Done
- Emphasis on full Product life-cycle, full feature life-cycle, and value realization
- Sprint Planning Topics 1-3 renamed to the Why, What, and How; Sprint up to 4 weeks rather than up to 1 month
- Possible additional review of the Increment and outcomes in a psychologically safer environment at the Sprint Retrospective
- More emphasis that the Increment is always Done, hence ‘Done Increment’ is superfluous
- Explicit about the malleability of the Product Goal (within reason)
- From the optimistic assumption of value delivery to an intentional *Focus* on value realization
- Ethos of built-in quality, clarity, data-informed decisions, intentional interactions, emergence (71), outcomes over outputs, pause & reflection, realizing value, understanding the problem or opportunity, fostering the climate/environment for a coherent Scrum adoption, and continuous improvement in a direction
- De-emphasis of the nebulous organization in order to attach the change to roles
- More intentional observance of the Scrum Values, considering the context

2.2 Appendix

Section 2: MORE executive SUCCESS extract

Title: MORE executive SUCCESS extract

Author: John Coleman

Source: (6)

License/Copyright: CC BY-NC-ND 4.0, © 2017-2025 Orderly Disruption Limited

Note: This section is included in its original, unaltered form with permission under the terms of the CC BY-NC-ND 4.0 license. No changes have been made.

2.2.1 The Adaptive Enterprise

It's difficult for an enterprise to be adaptive (80) without a climate where words and actions match. Over eighty engagement models were studied. Amongst those were scaling or descaling frameworks, and Product operating models, which can be useful for multi-Scrum-Team Products. Models range from going too far to not doing enough in helping the Product organization become more adaptive. There is no grand, universal truth or context-free 'Goldilocks zone.'

Of the engagement models studied, there are a number of notable contenders, including but not limited to Beyond Budgeting, Humanocracy, and Sociocracy, that, depending on the context, should be explored. Consider the combination with each other and with other approaches.

Beyond Budgeting Beyond Budgeting (15-28, 90-98, 103) is a management philosophy that rejects traditional, rigid annual budgeting in favor of a decentralized and adaptive approach to organizational control and performance management. It is built on 12 guiding principles-six focused on leadership and six on management processes-that promote decentralized decision-making, transparency, team autonomy, and a strong alignment with customer value.

Instead of fixed targets and detailed annual plans, Beyond Budgeting encourages dynamic goal-setting, continuous planning, and allocation informed by real-time needs, fostering adaptiveness and responsiveness in a rapidly changing business environment. This approach aims to empower teams, enhance innovation, and ensure organizations are better equipped to navigate uncertainty (72) and complexity (30-35). Beyond Budgeting is badly named (false assumption it's only about Finance) and well named at the same time (indeed going beyond budgeting).

Humanocracy Humanocracy (2), as defined by Gary Hamel, is a management model that replaces rigid hierarchies and centralized control with systems that maximize each person's contribution and creativity. In a humanocracy, organizations exist to serve and empower people, not just treat employees as resources for company goals.

It is built on principles like distributed ownership, meritocracy, openness, experimentation, and community, fostering autonomy and innovation. Authority is based on competence, and decision-making is decentralized to those closest to the work. Humanocracy prioritizes trust, engagement, and unleashing human potential over compliance and control, aiming to build resilient, innovative workplaces where employees drive meaningful change.

While models like Haier's Rendanheyi (56, 101) share values of decentralization and empowerment, humanocracy is a broader philosophy focused on replacing bureaucracy with people-centric principles that unlock collective capability and value.

Sociocracy Sociocracy (1,11-14) is a governance system that organizes people into self-managing (49) circles and makes decisions by consent, not majority vote. Developed by Gerard Endenburg (81) in the Netherlands in the 1970s, it ensures everyone affected by a decision has a voice, with proposals advancing unless a reasoned objection is raised. Guided by the principle of 'good enough for now, safe enough to try,' sociocracy distributes authority, promotes transparency, accountability, and continuous improvement, and fosters collaboration and shared ownership. Its principles have influenced models like Holacracy and self-managing teams.

The most established variant is the Sociocratic Circle-Organization Method (SCM), the original, formalized method. SCM uses semi-autonomous circles, double-linking (where two people attend two directly related circles to connect those circles), consent-based decision-making, and open elections for roles. This structure maintains both organizational efficiency and member equivalence, and has a well-documented track record in businesses, cooperatives, and schools in the Netherlands.

While newer variants like Sociocracy 3.0 (S3) offer more flexibility, SCM remains the most historically validated and widely documented form of sociocracy.

2.2.2 The Adaptive Executive or Board Member

MORE Executive SUCCESS identifies a number of opportunities for executives and board members:

- Acquire knowledge on stakeholders (including the customer) and their needs and limits, the work, how the work works, the waste, the anti-patterns, the problem space, the opportunities, the evidence that value can be harvested, behaviors, and habits
- Foster a humane performance climate and enable succession planning that protects and improves it
- Develop responsiveness and flow (68,69) across value networks
- Nurture emergence (71) and adaptiveness (80) in a direction with clarity
- Engage people, including customers and colleagues
- Foster effective and timely succession planning

There is abundant guidance for those from the organization's structural bottom, middle, and sides on how to improve adaptiveness (80). The executive level, however, is poorly served with guidance on timely humane effectiveness, customer interactions, and 'how the work works.' There is a misconception that hired change agents fill the gap alone, which is unrealistic because the organization owns the change.

Timely, humane effectiveness should permeate the entire corporate structure to gain its numerous benefits. Even organizations that have 'succeeded in change adoption' face hazards. People leave, other perspectives take hold, and corporate fads can unravel adaptiveness gains. Negative chaos could arise.

Many players and engagement models purportedly support executive adaptiveness, which is great because different organizational contexts require different approaches. But for all the resources available, the overall landscape of executive adaptiveness hasn't changed much in 25+ years.

Whether using tactics, strategies, methods, and frameworks or none, organizations should first embrace the ethos that underpins ambidexterity, humane effectiveness, adaptiveness, and timeliness at the top. Otherwise, executives and board members will continue to oversee 'change theater' and an incomplete patchwork of timely, humane, effective pockets within organizations.

Shining a Light on Executive Behavior Executive and board member posture or actions will influence the new behavior of others more than any of their words or directives. Nevertheless, it would be best to revise the questions asked to improve ambidexterity, humane effectiveness, adaptiveness, and timeliness.

Ambidexterity, humane effectiveness, adaptiveness, and timeliness require the eventual extinction of incoherent executive behavior. Examples of more helpful behaviors are accepting failure, seeking information before judging, giving opportunities to try something new to learn things, making it okay not to know, and helping people focus. There are some notable options for dealing with executive behavior.

Immunity To Change® {#immunity-to-change®} Lisa Laskow Lahey and Robert Kegan (principals at The Developmental Edge) created a change approach known as Immunity to Change® (3,4). People often know what to do, but they won't do it because of conflicting internal commitments. Metaphorically, people have 'one foot on the gas and one foot on the brake'.

Immunity to Change® is a framework for defining those 'hidden commitments' and 'limiting assumptions' that prevent people from changing and realizing their goals. The Immunity to Change® theory and map have helped countless professionals and organizations to unearth and move beyond the commitments preventing their professional and organizational growth.

Intent-Based Leadership® {#intent-based-leadership®} *Intent-Based Leadership® (IBL)* (7, 8, 9) *is a language teams use for high performance that replaces the programmed industrial-age language. IBL stresses the concept of intent from leaders and the team. It is based on the books Turn The Ship Around and Leadership is Language by L. David Marquet.*

One of the core beliefs is that leadership is not for the select few at the top. In highly effective organizations, there are leaders at every level. L. David Marquet molded the leadership he developed on the nuclear-powered submarine USS Santa Fe into a system called Intent-Based Leadership for your organization to implement to invite thinking and leadership at every level.

Intent-Based Leadership helps leaders build organizations where people are at their best because they have a sense of autonomy, tap their intrinsic motivation, feel listened to, and have a drive for excellence. They feel high levels of ownership and control, so they engage their hearts and heads. They gain psychological rewards as they see the fruits of their decisions and work. There is a bias for action, and teams are more agile and resilient because error creation and propagation are reduced.

The practice of stating intent allows teams to have distributed decision-making while maintaining unity of effort. The Intent-Based Leadership International (IBLI website) offers consulting, coaching, online courses, and books for leaders.

Section 3: Cynefin Framework Kind of Explanation unofficial & unauthorized

Title: Cynefin Framework Kind of Explanation unofficial & unauthorized
Source: [Link to original Cynefin wiki], [Link to this adaptation]
License: Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). © 2017-*2025 Cynefin.io.*
Disclaimer: No warranties are given. Use at your own risk.
This section is offered under the Attribution-ShareAlike 4.0 International license of Creative Commons.
By using this Cynefin Framework Kind of Explanation unofficial & unauthorized, you agree to the terms of the CC BY-SA 4.0 license.

2.2.3 Cynefin® {#cynefin®}

Cynefin® (30-35) offers a compass for leadership decision-making. It was popularized by the HBR article 'A Leader's Framework for Decision Making' by Dave Snowden and Mary Boone in 2007 and again in 'Managing complexity (and chaos) in a crisis - a field guide for decision makers inspired by the Cynefin framework', also known as the 'EU Field Guide.' Its premise is that one should act differently depending on the dynamics of the space. It is often oversimplified. A given problem could exist in all domains simultaneously, each having different aspects.

A phase shift refers to an often abrupt transition between domains, particularly from the ordered to

chaos, triggered when a system's constraints (rules, habits, boundaries, and feedback) are misaligned or collapse. It marks a fundamental change in system behavior where previous methods of control or understanding no longer work.

Not all aspects of Product development are complex. The Scrum Team, for a given situation, might need to consider a variety of phase shifts between:

- Ordered: Key idea: Stability, routine, best/good practice, expertise
  - Expertise is sufficient, and cause and effect are predictable or knowable
  - Response options not limited to: Apply best/good practices, follow rules, use expert analysis, do individual research
  - Metaphors: Hard to barely frozen ice cube, pleasant weather, or chess/sudoku
  - Nature example: A modern, climate-controlled glasshouse––predictable, controlled, planned growth
  - Product example: Resolving a tricky technical issue by consulting experts and analyzing logs
- Complex (30-35), where expertise is valuable but not enough, and one can only understand why things happened after the fact. Key idea: emergence, safe-to-fail experiments
  - Responses not limited to:
    * Encourage learning and adaptation
    * Trying several small, parallel, safe-to-fail experiments
    * Fostering fresh thinking through cognitive diversity and collaboration
    * Borrowing solutions from other places if they might help
    * Testing out smart guesses or informed hunches to see what works
  - This is all while following helpful guidelines that encourage good results to develop naturally
  - Metaphors: Flowing water, rainy weather, or poker
  - Nature example: Bramble thicket––everything is entangled, connections are unpredictable
  - Product example: Experimenting with different features or solutions informed by user feedback, e.g., A/B testing new Product ideas
- Chaotic:
  - Negative: Key idea: Destructive crisis, breakdown, urgent action
    * Responses not limited to: Take immediate action to restore order, prioritize safety, do something quickly without making matters worse
    * Metaphors: Shattering ice or uncontrolled explosion, toxic gas, tornado, earthquake, wildfire, or a riot in a stadium
    * Nature example: Natural disaster (e.g., tsunami)–sudden, destructive, unpredictable
    * Product example: Responding to a critical security breach by isolating systems and deploying emergency fixes
  - Positive: Key idea: Generative disruption, rapid innovation
    * Response options not limited to: Disrupt intentionally, encourage creativity, harness energy, e.g., hackathon, incubator, 'Innovation Friday'

        - Metaphors: Controlled explosion (steam engine), fireworks, or festival bonfire
        - Nature example: Forest fire clearing old growth for new plants–ecosystem renewal
        - Product example: Rapidly pivoting a Product during a market disruption to seize new opportunities (e.g., launching a feature in response to a competitor's move)

Liminality is an 'in-between' stage, like a threshold. The often less-sudden phase shifts happen in the liminals:

- In the liminal between complex and ordered, this is Scrum's default sweetspot:
    - Ordered-Complex:
        - From expert analysis to adaptive exploration
        - Responses not limited to: Relax some rules, introduce experimentation, prepare for emergence
        - Metaphors: A melting ice cube, cloudy weather, switching from chess to poker
        - Nature example: Seasonal thaw–rigid ice giving way to flowing streams and new growth
        - Product example: When a routine process stops working, encourage the team to try different approaches
    - Complex-Ordered:
        - Responses not limited to: Turn creative discoveries into expert routines; stabilize innovation, observe and codify successful patterns; transition to standardization
        - Metaphors: Slush (between ice and water), fog lifting after rain, switching from poker to chess
        - Nature example: River delta forming channels–from unpredictable to stable flows
        - Product example: Taking a successful experimental feature and turning it into a documented, repeatable process
- In the liminal between complex and chaotic:
    - Complex–Chaotic (positive):
        - A situation where constraints need to be relaxed to create time and space for innovation or invention. Key idea: The edge of creativity, risk, and innovation
        - Responses not limited to: Loosen constraints, encourage experimentation, seek breakthrough ideas
        - Metaphors: Boiling water (on the edge of steam), thunderstorm breaking out, Improvizational theater, or jazz jam session
        - Nature example: Volcano creating new land–creative transformation at the edge of chaos
        - Product example: Running a high-risk innovation hackathon to generate disruptive ideas
    - Complex–Chaotic (negative):
        - Key idea: A destructive move into crisis
        - Responses not limited to: Rapidly re-impose constraints, stabilize the situation, prevent further breakdown
        - Metaphors: Exploding pressure cooker, sudden tornado or flash flood, game pieces thrown in anger, game board upended

        - Nature example: Sudden landslide–loss of structure, destructive transition
        - Product example: Failed Product launch confusion, and urgent need to regain control
    - Chaotic–Complex: Getting out of chaotic––regrouping
        - Response options not limited to: Sense emerging order, start probing, encourage collaboration, and pattern recognition
        - Metaphors: Steam condensing to water, calm after a hurricane, restarting a sports game after a storm
        - Nature example: Pioneer species colonizing after fire–new growth after disturbance
        - Product example: After a crisis, regrouping the team to experiment with new ways of working or new Product directions

- Aporia (paradoxical liminal): sitting with paradox for new insight, perhaps after realizing the situation at hand was not as it seemed

    - Response options: Hold ambiguity, encourage reflection, allow new understanding to emerge
    - Metaphors: Triple point (solid, liquid, gas coexist), standing in the eye of a storm, solving a riddle
    - Nature example: Estuary where river, sea, and land meet–all states and possibilities coexist
    - Product example: The team is stuck between conflicting strategies or visions and should briefly pause to reflect and realign.

- A rarely considered phase shift due to difficulty level: Chaotic-Orderly liminal

    - Response options: Impose strong constraints, re-establish rules and structure
    - Metaphors: Ice rapidly refreezing, cold snap after a storm, referee is successfully strict after chaos
    - Nature example: A Dam is successfully built after a flood––a wild river suddenly contained and controlled
    - Product example:
        - After a major production outage or Product crisis, a cross-functional crisis team rapidly stabilizes the situation with clear, minimal rules and temporary protocols
        - Once the immediate danger is past, these are iteratively refined and formalized into sustainable, balanced processes, avoiding overcorrection or excessive bureaucracy

One phase shift is particularly sudden and negative, that is, the ordered-chaotic liminal:

- Response options: Recognize fragility and over-confidence, act quickly to restore boundaries and safety
- Metaphors: Ice cracking into shards, sudden and violent hailstorm, game rules suddenly thrown out
- Nature example: Frozen lake breaking up in spring––stable surface suddenly shattering
- Product example: A stable Product process suddenly breaks down due to an unexpected event (e.g., major outage)

Section 4: Emergent Strategy

Author: Roger L. Martin, Tom Gilb

Source: (41-48)
Copyright: All rights reserved. Adapted

2.2.4 Emergent Strategy

Strategy is not limited by scale; if it exists, it should be clearly articulated at the corporate, business unit, or Product level and remain coherent and integrated across these levels. Crucially, strategy should distinguish between ends (quantified, Stakeholder-valued outcomes) and means (initiatives or activities).

Drawing from and adapting Roger L. Martin's work (41) and Tom Gilb's work (43-48), strategy is about making integrated, explicit choices- deciding what and what not to pursue from a well-defined, measurable winning aspiration, not just a broad mission or vision. Effective strategy answers:

- Where will we play?
- How will we win ethically (57) and sustainably, balancing a multiplicity of expectations and limits?
- What capabilities and systems must be in place?
- What else must be true for this strategy to succeed?

For situations where expertise alone is sufficient (or perhaps bordering on being sufficient), to ensure strategy is iterative, actionable, and value-focused:

- Iteratively quantify and manage Stakeholder value, multiple impacts or side effects, risks, and trade-offs:
  - Identify all critical Stakeholders (including but not limited to customers) and define their value objectives in measurable terms (e.g., 'reduce new-user onboarding time from 5-10 to 2-4 days').
  - Explicitly quantify trade-offs and constraints, and revisit as new information emerges.
  - Use integrative thinking to resolve tensions creatively.
- Co-create and prioritize collaboratively:
  - Develop the strategy by blending top-down and bottom-up insights and lateral collaboration.
  - Use structured workshops and feedback loops to foster alignment and adaptability, and continuously reprioritize unstarted work.
- Deliver value incrementally and measure results:
  - Iteratively break down strategic aspirations into small, prioritized, measurable increments.
  - Deliver value in short cycles (e.g., Sprints or weeks), measuring actual outcomes and side effects against original quantified objectives.
  - Use regular reviews to adjust informed by real-world feedback.
- Enable emergence:
  - Allow strategy to evolve in response to new data and Stakeholder (including but not limited to user) feedback, within a framework of clear, quantified objectives, measurable trends, and regular risk/benefit reassessment.
  - Make course corrections rapidly and transparently as reality unfolds.

- Ensure strategy and strategy deployment are outcome-oriented and focused (deciding what to and what not to work on). Distinguish between:
  - Strategy including the intent, rationale, goals, and anti-goals (the *what* and *why*),
  - Strategy deployment: the operationalization of the strategy, iterative sequencing or decomposition of integrated choices for the strategy, usually in small outcome-oriented slices of the *what* and *why*,
  - Outcome-oriented, focused Product Backlog Items (smaller slices for *whom*), and
  - Lists of activities or initiatives (the 'what we'll do' or *how*).
- Avoid mistaking a collection of projects for a coherent, value-driven strategy.

For situations where expertise is valuable yet insufficient, cause and effect are only coherent in retrospect, and uncertainty needs to be embraced, Scrum Teams and Stakeholders need to:

- Embrace the messiness of less structured and emergent outcome-oriented work in a direction of travel.
- Consider that detailed, long-term plans are ineffective. Instead, organizations should focus on creating conditions where useful patterns and innovations can emerge from interactions within the system.
- Instead of trying one idea at a time and sticking to what worked before, Scrum Teams should consider several small, parallel safe-to-fail experiments at once to see what happens and learn from what emerges.
- Foster a climate for creative exploration, innovation, and evolution from the present. Create processes and environments where people can connect novel ideas, learnings, informed hunches, and learn from each other, rather than imposing uniformity or rigid KPIs.
- Response options are not restricted to:
  - Map what is already known and understand the system's evolutionary potential before attempting change
  - Foster self-organization
  - Run safe-to-fail experiments (probes)–probes should be small, parallel, and designed so that failure is survivable and informative
  - Seek fresh thinking
  - Try solutions for different problems for the current situation at hand
  - Test educated hunches
  - Observe what emerges, and amplify successful patterns while dampening or stopping those that don't work
  - Innovation is important, but proven solutions should be reused for recurring problems
  - Continuously sense-make
  - Perform narrative capture
- Metaphor: The role of leaders is to actively prepare and manage the soil, boundaries, and conditions (the 'substrate') to encourage the growth of healthy plants (emergent solutions). This includes metaphorically weeding, pruning, and shaping the environment, not just passively waiting for results.

Generally, extrinsic motivation rewards should be avoided due to the 'cobra effect' (104) unless they are coherent with Beyond Budgeting. Equally, individual or team performance should be uncoupled from results as results might have been delivered, but in what way were they delivered, with what side effects, and what impact did delivery have on team morale, etc?

Nevertheless:

- There is disagreement in peer-reviewed papers (105-108) and a foundational non-peer-reviewed paper (109) on whether quantifying Stakeholder expectations, Stakeholder limits, or goals is helpful or unhelpful and whether it reduces intrinsic motivation.
- Consider the context. Also, consider whether quantification supports autonomy and meaning or imposes controlling constraints.
- For now, this document prefers to err on the side of clarification and shared understanding of an idea, quantifying Stakeholder expectations, Stakeholder limits, and the direction of travel, supported by high-quality and accurate storytelling narratives (more stories like this, fewer stories like that).

An Emergent Strategy is supported by an emergent outcome-oriented roadmap, which can range from the Sprint Goal to the Product Vision and beyond. Emergent Strategy Deployment (120-123) should not be confused with Emergent Strategy. Vector change models (30-35, 54), Product Operating Models (113-119), scaling and descaling models (134-147), and emergent goal-oriented models (120-133) can be highly beneficial for Emergent Strategy Deployment. Err on the side of models coherent with vector-change, e.g., direction of travel over fixed goals.

Emergent strategy deployment involves allowing plans and actions to develop naturally as the Scrum Team and Stakeholders respond to real-world changes. Instead of following a fixed path, they pay attention to what is happening around them and make adjustments as they go. Over time, the steps taken form a pattern that becomes the actual strategy, even if it differs from what was initially intended.

3

3.1 Attribution for the Scrum Guide Expansion Pack Collection

This collection was written and compiled by *Ralph Jocham, John Coleman, and Jeff Sutherland*. Each section is individually attributed above and retains its original license. The collection as a whole is for informational purposes; please respect the license terms of each section.

4

4.1 References

1. *Rau, T. (2022) Sociocracy - Basic Concepts and principles, Sociocracy For All. At: `https://www.sociocracyforall.org/sociocracy/` (Accessed: April 5, 2023).*
2. *Hamel, G. and Zanini, M. (2023) Humanocracy. At: `https://www.humanocracy.com/` (Accessed: April 5, 2023).*
3. *Kegan, R. and Laskow Lahey, L. (2019) An everyone culture, The Developmental Edge. At: `https://developmentaledge.com/an-everyone-culture/` (Accessed: April 4, 2023).*

4. *Laskow Lahey, L. and Kegan, R. (2023) News & thinking, The Developmental Edge. At: https://developmentaledge.com/newsthinking/#methodologies (Accessed: April 3, 2023).*
5. *Moore, G.A., 1991. Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers. New York: Harper Business.*
6. Coleman, J., (2025) MORE executive SUCCESS. Unpublished.
7. *Marquet, L. D. (2013) Turn the Ship Around! A True Story of Turning Followers into Leaders. Portfolio.*
8. *Marquet, L.D. (2021) Leadership is language: The hidden power of what you say and what you don't. Nakskov, Denmark: Nota.*
9. *Marquet, L. D. (2021) Based Leadership® International with L. David Marquet - IBLI. At:https://davidmarquet.com/ (Accessed: April 5, 2023).*
10. *Rau, T.J. and Koch-Gonzalez, J. (2018) Many voices one song: Shared power with sociocracy. Amherst, MA: Sociocracy for All.*
11. *Buck, J. & Endenburg, G. (2012) The creative forces of self-organization. Sociocratic Center.*
12. *Buck, J. & Villines, S. (2017) We the people: Consenting to a deeper democracy. 2nd edn. Sociocracy.info Press.*
13. *Endenburg, G. (1998) Sociocracy: The organization of decision-making. Delft: Eburon Publishers.*
14. *Priest, J. & Bockelbrink, B. (2018) Sociocracy 3.0 – The practical guide. Available at: https://sociocracy30.org/ (Accessed: 17 May 2025).*
15. *Bogsnes, B. (2023) This is beyond budgeting: A guide to more adaptive and human organizations. Hoboken, NJ: John Wiley & Sons, Inc.*
16. *Bogsnes, B. (2023) Beyond budgeting at 25 - bbrt.org, Beyond Budgeting Round Table. At: https://bbrt.org/wp-content/uploads/bb-white-paper_a.pdf (Accessed: April 7, 2023).*
17. *Olesen, A. (2016) Beyond budgeting: Principle 1 - purpose, YouTube. At: https://youtu.be/_9ZW2NjyFxE (Accessed: April 7, 2023).*
18. *Larsson, D. (2016) Beyond budgeting: Principle 2 - values, YouTube. At: https://youtu.be/pl1BPrITbm4 (Accessed: April 7, 2023).*
19. *Player, S. (2016) Beyond budgeting: Principle 3 - transparency, YouTube. At: https://youtu.be/Mb7K8App2vw (Accessed: April 7, 2023).*
20. *Röösli, F. (2016) Beyond budgeting: Principle 4 - Organization, YouTube. At: https://youtu.be/i8HIgc8OZYM (Accessed: April 7, 2023).*
21. *Larsson, D. (2016) Beyond budgeting: Principle 5 - autonomy, YouTube. At: https://youtu.be/ipnjHtXYi-g (Accessed: April 7, 2023).*
22. *Player, S. (2016) Beyond budgeting: Principle 6 - customers, YouTube. At: https://youtu.be/_6fut4R_wVw (Accessed: April 7, 2023).*
23. *Bogsnes, B. (2016) Beyond budgeting: Principle 7 - rhythm, YouTube. At: https://youtu.be/rb_NsnPNIQQ (Accessed: April 7, 2023).*
24. *Röösli, F. (2016) Beyond budgeting: Principle 8 - targets, YouTube. At: https://youtu.be/up3mp7jN6XU (Accessed: April 7, 2023).*
25. *Player, S. (2016) Beyond budgeting: Principle 9 - plans and forecasts, YouTube. At: https://youtu.be/OWM7FUuXejI (Accessed: April 7, 2023).*
26. *Olesen, A. (2016) Beyond budgeting: Principle 10 - resource allocation, YouTube. At: https://youtu.be/mPCYHmvi_b8 (Accessed: April 7, 2023).*

27. *Bogsnes, B. (2016) Beyond budgeting: Principle 11 - performance evaluation, YouTube. At: https://youtu.be/RfPVtG2B27E (Accessed: April 7, 2023).*
28. *Röösli, F. (2016) Beyond budgeting: Principle 12 - rewards, YouTube. At: https://youtu.be/ETU5TzNYiC0 (Accessed: April 7, 2023).*
29. *Takeuchi, H. and Nonaka, I. (2014) The new new product development game, Harvard Business Review. At: https://hbr.org/1986/01/the-new-new-product-development-game (Accessed: 21 January 2024).*
30. *Cynefin.io, V. (2022) Cynefin wiki, Cynefin.io. Cynefin.io. At: https://cynefin.io/ (Accessed: April 4, 2023).*
31. *Rancati, A. and Snowden, D. (2021) Managing complexity (and chaos) in a crisis - a field guide for decision makers inspired by the Cynefin framework. Luxembourg, Belgium: Publications Office of the European Union.*
32. *Snowden, D. et al. (2022) Cynefin® weaving sense-making into the fabric of our world. 2nd edn. Edited by R. Greenberg and B. Bertsch. Singapore, Singapore: Cognitive Edge - The Cynefin Co.*
33. *Snowden, D. (2023) Cynefin St David's 2023 1 of 2, Cynefin Co. https://thecynefin.co/cynefin-st-davids-2023 (Accessed: April 20, 2023).*
34. *Snowden, D. (2023) Managing for emergence through abduction, The Cynefin Co. At: https://thecynefin.co/managing-for-emergence/ (Accessed: June 24, 2023).*
35. *Snowden, D. and Smith, N. (2023) Leadership discussion: Dave and Natalie - the Cynefin co, YouTube. At: https://youtu.be/WcPZ8ybDF0w (Accessed: April 7, 2023).*
36. *Langton, C.G. (ed.) (1989) Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems, Los Alamos, New Mexico, September 1987. Santa Fe Institute Studies in the Sciences of Complexity, vol. VI. Redwood City, CA: Addison-Wesley.*
37. *Langton, C.G. (1989) 'Life at the edge of chaos', in Langton, C.G. (ed.) Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems. Santa Fe Institute Studies in the Sciences of Complexity, vol. VI. Redwood City, CA: Addison-Wesley, pp. 41–91.*
38. *Wolfram, S. (2002) A new kind of science. Champaign, IL: Wolfram Media.*
39. *Alexander, C. (1979) The timeless way of building. New York: Oxford University Press.*
40. *Schwaber, K. & Sutherland, J. (2020) The Scrum Guide: The definitive guide to Scrum: The rules of the game. Available at: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf (Accessed: 17 May 2025)*
41. *Martin, R.L. (2022) A new way to think your guide to Superior Management Effectiveness. Boston, MA, MA, USA: Harvard Business Review Press.*
42. *Gilb, T. & Graham, D. (1993) Software Inspection. Harlow: Addison-Wesley.*
43. *Gilb, T. (1988) 'Deeper perspectives on evolutionary delivery, in Principles of Software Engineering Management. Wokingham: Addison-Wesley, pp. [chapter 15]. Also available at: https://bit.ly/TomGilbEvo.*
44. *Gilb, Tom & Maier, Mark. (2005). Managing Priorities: A Key to Systematic Decision Making. INCOSE International Symposium. 15. 10.1002/j.2334-5837.2005.tb00782.x. Also available at: https://bit.ly/TomGilbPriorities.*
45. *Gilb, T. (1988) 'Deeper perspectives on evolutionary delivery', in Principles of Software Engineering Management. Wokingham: Addison-Wesley, pp. [chapter 15].*

46. *Gilb, T. (2005) Competitive Engineering: A Handbook for Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage. Oxford: Elsevier Butterworth-Heinemann. Also available at: https://bit.ly/TomGilbCompEng.*
47. *Gilb, T. (2009) 'Agile specification quality control: Shifting emphasis from cleanup to sampling defects', Testing Experience, March. Available at: https://www.researchgate.net/publication/294196272_Agile_specification_quality_control [Accessed: 17 May 2025].*
48. *Gilb, T. & Gilb, K. (1989) 'The McDonnell-Douglas case study of SQC and engineering improvement: Case DAC Inspection 1988–89'. Available at: https://bit.ly/TomGilbMcDonnell-Douglas [Accessed: 17 May 2025].*
49. *LeSS.works (n.d.) Self-managing teams. Available at: https://less.works/less/management/self-managing-teams (Accessed: 17 May 2025).*
50. *Gothelf, J. & Seiden, J. (2021) Lean UX: Designing great products with agile teams. 3rd edn. Sebastopol, CA: O'Reilly Media*
51. *Torres, T. (2021) Continuous discovery habits: Discover products that create customer value and business value. North Charleston, SC: Product Talk*
52. *Scrum.org (2025) Scrum Hexis. Available at: https://thecynefin.co/product/hexi-scrumorg/?srsltid=AfmBOorcohLYeVy0qBsQFI6mK_bZtJA_uGC6hPL2BdptiTwNmMwpKTQv (Accessed: 17 May 2025).*
53. *Sutherland, J., Coplien, J.O., Heasman, L., den Hollander, M., Ramos, C. and The Scrum Patterns Group (2019) A Scrum Book: The Spirit of the Game. Raleigh, NC: Pragmatic Press.*
    Members of The Scrum Patterns Group: Vervloed, E., Harrison, N., Harada, K., Yoder, J., Kim, J., O'Callaghan, A., Beedle, M., Bjørnvig, G., Friis, D., Reijonen, V., Benefield, G., Østergaard, J., Eloranta, V.-P., Leonard, E. & Aguiar, A.
54. *Snowden, D. (2025) 'Estuarine mapping first edition', The Cynefin Co, 22 April. Available at: https://thecynefin.co/estuarine-mapping/ (Accessed: 8 June 2025)*
55. *Ackoff, R.L. (1999) Ackoff's Best: His Classic Writings on Management. New York: John Wiley & Sons.*
56. *Fischer, B., Minnaar, J., Moehrle, M., & Cornuel, E. (2020) RenDanHeYi: Pioneering the Quantum Organisation. EFMD Global Focus, Special Supplement. Available at: https://bit.ly/RenDanHeYi [Accessed 27 May 2025]*
57. *Blackburn, S. (2003) Ethics: A Very Short Introduction. Oxford: Oxford University Press.*
58. *Mayer, T. (2025) A Simple Guide to Scrum. [Online]. Available at: https://scrum.academy/guide/ (Accessed: 17 May 2025)*
59. *Ohno, T. (1988) Toyota Production System: Beyond Large-Scale Production. Portland, OR: Productivity Press.*
60. *Toyota Motor Corporation (2024) Toyota Production System. Available at: https://global.toyota/en/company/vision-and-philosophy/production-system/index.html (Accessed: 17 May 2025).*
61. *Hounshell, D.A. & Smith, J.K. (1988) Science and Corporate Strategy: DuPont R&D, 1902–1980. Cambridge: Cambridge University Press.*
62. *Schwaber, K. and Sutherland, J. (1995) 'SCRUM Development Process', OOPSLA Business Object Design and Implementation Workshop. Austin, Texas, October 1995. Available at: http://jeffsutherland.org/oopsla/schwapub.pdf (Accessed: 17 May 2025).*

63. *Womack, J.P. and Jones, D.T. (1996) Lean Thinking: Banish Waste and Create Wealth in Your Corporation. New York: Simon & Schuster.*
64. *Thurlow, N., Turner, J.R. and Podder, A. (2020) The Flow System: The Evolution of Agile and Lean Thinking in an Age of Complexity. Flow Consortium. Available at: https://flowguides.org/Flow_Guide.pdf (Accessed: 17 May 2025).*
65. *Felderer, M. and Travassos, G.H. (2020) 'The Evolution of Empirical Methods in Software Engineering'. Available at: https://arxiv.org/pdf/1912.11512.pdf (Accessed: 17 May 2025).*
66. *Creative Wisdom (n.d.) 'Abduction, Deduction and Induction'. Available at: https://www.creative-wisdom.com/teaching/WBI/abduction5.pdf (Accessed: 17 May 2025).*
67. *Campbell, J. (2025) 'Empiricism', EBSCO Research Starters. Available at: https://www.ebsco.com/research-starters/religion-and-philosophy/empiricism (Accessed: 17 May 2025)*
68. *Kanban Guides (2025) Available at: https://kanbanguides.org (Accessed: 17 May 2025)*
69. *Scrum.org et al. (2021) The Kanban Guide for Scrum Teams. Available at: https://www.scrum.org/resources/kanban-guide-scrum-teams (Accessed: 17 May 2025)*
70. *Csíkszentmihályi, M. (1990) Flow: The Psychology of Optimal Experience. New York: Harper & Row*
71. *Templeton Foundation (2023) 'What Is Emergence?' John Templeton Foundation. Available at: https://www.templeton.org/news/what-is-emergence (Accessed: 17 May 2025).*
72. *van der Bles, A.M., van der Linden, S., Freeman, A.L.J. and Spiegelhalter, D.J. (2019) 'Communicating uncertainty about facts, numbers and science', Royal Society Open Science, 6(5), 181870. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6549952/ (Accessed: 17 May 2025).*
73. *Morieux, Y. (2015) How too many rules at work keep you from getting things done: Yves Morieux: Ted Talks, YouTube. At: https://youtu.be/t NoFstCmQ (April 3, 2023).*
74. *Holland, J.H. (1992) Complex Adaptive Systems. Daedalus, 121(1), pp. 17–30. Available at: https://www.jstor.org/stable/20025416 (Accessed: 17 May 2025).*
75. *Axelrod, R. and Cohen, M.D. (2000) Harnessing Complexity: Organizational Implications of a Scientific Frontier. New York: Free Press.*
76. *Juarrero, A. (1999) Dynamics in Action: Intentional Behavior as a Complex System. Cambridge, MA: MIT Press.*
77. *Snowden, D.J. and Boone, M.E. (2007) 'A leader's framework for decision making', Harvard Business Review, 85(11), pp. 68–76. Available at: https://hbr.org/2007/11/a-leaders-framework-for-decision-making (Accessed: 17 May 2025)*
78. *Dictionary Marketing (2024) 'B2B2B'. Available at: https://dictionarymarketing.com/definition/b2b2b/ (Accessed: 17 May 2025).*
79. *NetSuite (2023) 'What Is Business to Business to Consumer (B2B2C)?' Available at: https://www.netsuite.com/portal/resource/articles/ecommerce/b2b2c.shtml (Accessed: 17 May 2025).*
80. *LeSS (n.d.) 'Why LeSS? Achieving adaptiveness'. Available at: https://less.works/less/framework/why-less (Accessed: 17 May 2025).*
81. *Sociocracy For All (n.d.) 'Gerard Endenburg: founder of Sociocratic Circle Method and pioneer of self-management'. Available at: https://www.sociocracyforall.org/gerard-endenburg-founder-of-sociocratic (Accessed: 18 May 2025).*
82. *Patton, J. and Economy, P. (2014) User Story Mapping: Discover the Whole Story, Build the Right Product. Sebastopol, CA: O'Reilly Media.*

83. *Kotter, J.P., 1996. Leading Change. Boston: Harvard Business School Press.*
84. *'Genchi Genbutsu' (2024) Wikipedia. Available at: https://en.wikipedia.org/wiki/Genchi_genbutsu (Accessed: 18 May 2025).*
85. *ScrumPlop, n.d. Illigitimus Non Interruptis. The Scrum Book: The Spirit of the Game. Available at: https://sites.google.com/a/scrumplop.org/published-patterns/product-organization-pattern-languag illegitimus-non-interruptus [Accessed: 18 May 2025].*
86. *Cagan, M., 2018. Inspired: How to Create Tech Products Customers Love. 2nd ed. Hoboken, NJ: Wiley.*
87. *Cagan, M. & Jones, C., 2020. Empowered: Ordinary People, Extraordinary Products. Hoboken, NJ: Wiley.*
88. *Cagan, M., 2024. Transformed: Moving to the Product Operating Model. Hoboken, NJ: Wiley.*
89. *Schwaber, K. (2023) 'Scrum Guide', Ken Schwaber's Blog, 25 September. Available at: https://kenschwaber.wordpress.com/2023/09/25/scrum-guide/ (Accessed: 20 May 2025).*
90. *Future Ready: How to Master Business Forecasting*
    Morlidge, S. & Player, S., 2010. Future Ready: How to Master Business Forecasting. Chichester: John Wiley & Sons.
91. *The Little Book of Beyond Budgeting*
    Morlidge, S., 2024. The Little Book of Beyond Budgeting: A New Management Model for Organisations (Second Edition) [Beyond Books Press]
92. *The Little (Illustrated) Book of Operational Forecasting*
    Morlidge, S., 2019. The Little (Illustrated) Book of Operational Forecasting. [Troubador].
93. *Present Sense*
    Morlidge, S., 2019. Present Sense. [Troubador].
94. *Zen and the Art of Organising Work*
    Morlidge, S., 2021. Zen and the Art of Organising Work. [Troubador].
95. *Cost Matters*
    Morlidge, S., 2023. Cost Matters. [Beyond Books Press].
96. *Beyond Budgeting i praktiken Fahlén, K., 2016. Beyond Budgeting i praktiken. Stockholm: Liber.*
97. *Fahlén, K., 2018. Dynamic Management Strategy: A guide to management innovation and competitive advantage. Gothenburg: BAS*
98. *Bogsnes, B., 2016. Implementing Beyond Budgeting: Unlocking the Performance Potential. 2nd ed. Chichester: John Wiley & Sons.*
99. *Boyd, J.R. (1995–1996) The Essence of Winning and Losing. Unpublished briefing slides. Note: Boyd's OODA was primarily disseminated through military briefings and unpublished manuscripts. His final conceptualization appears in The Essence of Winning and Losing, which emphasizes nonlinear decision-making and adaptation in complex environments.*
100. *Turner, J.R., Thurlow, N. and Rivera, B. (2019) The Flow System Guide. Available at: https://flowguides.org/Flow_Guide.pdf (Accessed: 24 May 2025). Summary: This guide integrates Boyd's OODA with complexity theory and agile practices, framing it as a dynamic, non-linear decision-making process for organizational flow.*
101. *Williamson, P.J. & Yin, E. (2018) 'Management Innovation Made in China: Haier's Rendanheyi', California Management Review, 61(1), pp. 71-93.*

102. *Richards, C. (2004) Certain to Win: The Strategy of John Boyd, Applied to Business. Bloomington, IN: Xlibris*
103. *Becker, S et al (co-author) The Viable Map Workbook 2023 [Beyond Books Press]*
104. *Frey, B.S. and Jegen, R. (2001) 'Motivation crowding theory', Journal of Economic Surveys, 15(5), pp. 589–611.*
105. *Cameron, J., Banko, K.M. and Pierce, W.D. (2001) 'Pervasive negative effects of rewards on intrinsic motivation: The myth continues', The Behavior Analyst, 24(1), pp. 1–44.*
106. *Deci, E.L., Koestner, R. and Ryan, R.M. (1999) 'A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation', Psychological Bulletin, 125(6), pp. 627–668.*
107. *Ryan, R.M. and Deci, E.L. (2000) 'Intrinsic and extrinsic motivations: Classic definitions and new directions', Contemporary Educational Psychology, 25(1), pp. 54–67.*
108. *Sandel, M.J. (2012) What money can't buy: The moral limits of markets. London: Allen Lane.*
109. *Kohn, A. (1993) 'Why incentive plans cannot work', Harvard Business Review, 71(5), pp. 54–63.*
110. *Fuzzy Business: How to be roughly right rather than precisely wrong (unpublished).*
111. *Lewis, R. (2023) An operating model for business agility: Agile for managers of the digital age. Independently published.*
112. *less.works (n.d.) Technical Excellence. Available at: `https://less.works/less/technical-excellence` (Accessed: 7 June 2025)*
113. *Cagan, M. (2024) Transformed: Moving to the Product Operating Model. Hoboken, NJ: Wiley.*
114. *Cagan, M. (2025) 'The Product Operating Model', Silicon Valley Product Group, 17 March. Available at: `https://www.svpg.com/the-product-operating-model/` (Accessed: 8 June 2025).*
115. *Cagan, M. (n.d.) 'The Product Operating Model: An Introduction', Silicon Valley Product Group. Available at: `https://www.svpg.com/the-product-operating-model-an-introduction/` (Accessed: 8 June 2025)*
116. *Scrum.org (2025) 'The Agile Product Operating Model', Scrum.org, 1 May. Available at: `https://www.scrum.org/resources/agile-product-operating-model` (Accessed: 8 June 2025).*
117. *Scrum.org (2025) 'Agile Product Operating Model State of Play - Part 1 - Fundamentals', Scrum.org, 12 May. Available at: `https://www.scrum.org/resources/blog/agile-product-operating-model-state-play-` (Accessed: 8 June 2025).*
118. *Scrum.org (2024) 'Project to Product and the Agile Product Operating Model', Scrum.org, 7 November. Available at: `https://www.scrum.org/resources/blog/project-product-and-agile-product-operating-mode` (Accessed: 8 June 2025).*
119. *Scrum.org (2024) Moving to an Agile Product Operating Model [PDF]. Available at: `https://www.scrum.org/resources/moving-agile-product-operating-model-evidence-based-approach-delivering-pro` or `https://bit.ly/SDOAPOM`. (Accessed: 8 June 2025)*
120. *Scotland, K. (2023) Why strategy deployment? Here are three great reasons, AvailAgility. At: `https://availagility.co.uk/2023/02/16/why-strategy-deployment-here-are-three-great-reasons/` (Accessed: April 3, 2023).*
121. *Scotland, K. (2019) Deploying strategies as choices, AvailAgility. At: `https://availagility.co.uk/2019/02/08/deploying-strategies-as-choices/` (Accessed: April 3, 2023).*
122. *Scotland, K. (2017) Strategy deployment and playing to win, AvailAgility. At: `https://availagility.co.uk/2017/07/14/strategy-deployment-and-playing-to-win/` (Accessed: April 3, 2023).*

123. *Scotland, K. (2017) A strategy deployment cadence, AvailAgility. At: https://availagility.co.uk/2017/09/06/a-strategy-deployment-cadence/ (Accessed: April 3, 2023).*
124. *Scotland, K. (2022) The ultimate X-matrix for your agile transformation is here, AvailAgility. At: https://availagility.co.uk/2022/11/03/the-ultimate-x-matrix-for-youragile-transformation-is-here (Accessed: April 5, 2023).*
125. *Krebs, J. (2023) Agile kata pro, Agile Kata Pro. At: https://agilekata.pro/ (Accessed: April 4, 2023).*
126. *Doerr, J. (2023) OKRs 101, What Matters. At: https://www.whatmatters.com/get-started/ (Accessed: April 4, 2023).*
127. *Wodtke, C. (2021) Radical focus achieving your most important goals with objectives and key results–. Palo Alto, CA: Cucina Media.*
128. *Gothelf, J. & Seiden, J. (2024) Who Does What By How Much?: A Practical Guide to Customer-Centric OKRs. New York: Sense & Respond Press.*
129. *Appelo, J. (2023) Sometimes, you *don't* want focus, unFIX. At: https://unfix.com/blog/sometimes-you-dont-wa (Accessed: 14 January 2024).*
130. *Appelo, J. (2023) Bets and objectives, unFIX. At: https://unfix.com/bets-and-objectives (Accessed: 14 January 2024).*
131. *McChesney, C. (2023) The 4 disciplines of execution (new), FranklinCovey. At: https://www.franklincovey.com/the-4-disciplines/ (Accessed: April 4, 2023).*
132. *Scrum.org (2024) Evidence-Based Management (EBM) Framework, Scrum.org. Available at: https://www.scrum.org/resources/evidence-based-management. (Accessed: 8 June 2025).*
133. *Burrows, M. (2023) Home: Agendashift™, Agendashift. At: https://www.agendashift.com/ (Accessed: April 4, 2023).*
134. *Kniberg, H. and Ivarsson, A. (2012) Scaling at Spotify, Crisp. At: https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf (Accessed: April 5, 2023).*
135. *Ambler, S.W. and Lines, M. (2023) Disciplined Agile® Toolkit - Project Management Institute, PMI. At: https://www.pmi.org/disciplined-agile/ (Accessed: April 5, 2023).*
136. *Leffingwell, D. and Knaster, R. (2023) Safe 6.0 framework, Scaled Agile Framework. At: https://www.scaledagileframework.com/ (Accessed: April 5, 2023).*
137. *Sutherland, J. (2021) Scrum@Scale - the scaling framework created by dr. Jeff Sutherland, Scrum@Scale Framework. At: https://www.scrumatscale.com/ (Accessed: April 5, 2023).*
138. *Skelton, M. and Pais, M. (2023) Team topologies, Team Topologies. At: https://teamtopologies.com/ (Accessed: April 5, 2023).*
139. *Appelo, J. (2023) Versatile Organization Design, unFIX. At: https://unfix.com/ (Accessed: April 5, 2023).*
140. *Merel, P. (2023) Xscale Alliance, XSCALE Alliance. At: https://xscalealliance.org/#manifesto (Accessed: April 5, 2023).*
141. *Schwaber, K. et al. (2021) Online nexus guide, Scrum.org. At: https://www.scrum.org/resources/online-nexus-guide (Accessed: April 5, 2023).*
142. *Quartel, R. et al. (2024) FaST guide, Fluid Scaling Technology. At: https://www.fastagile.io/ (Accessed: December 6, 2023).*
143. *Ramos, C. and Pavlichenko, I. (2023) Creating agile organizations, Creating Agile Organizations. At:*

https://creatingagileorganizations.com/ (Accessed: April 15, 2023).

144. *Larman, C. & Vodde, B. (2025) LeSS (Large-Scale Scrum) Framework. Available at: https://less.works/less/framework (Accessed: 8 June 2025)*
145. *Flight Levels GmbH (2025) Flight Levels Framework. Available at: https://www.flightlevels.io/what-is-flight-levels/ (Accessed: 8 June 2025).*
146. *Krivitsky, A. and Flemm, R. (2022) Org topologies, Org Topologies. At: https://www.orgtopologies.com/ (Accessed: April 4, 2023).*
147. *Singh, P. (2023) Scaling Simplified: A Practitioner's Guide to Scaling Flow. Florida: Self-published. Available at: https://leanpub.com/scalingsimplified (Accessed: 8 June 2025)*
148. *Davies, Dan. (2025) The Unaccountability Machine: Why Big Systems Make Terrible Decisions—and How the World Lost Its Mind. London: Profile Books Ltd. (Paperback edition).*
149. *Stripe (2025) 'Sir Jony Ive and Patrick Collison Fireside Chat | Stripe Sessions 2025', YouTube video, 8 May. Available at: https://youtu.be/wLb9g_8r-mE?si=1rEJxU0sxixvblQ3&t=1390 (Accessed: 8 June 2025)*