Agile Base Patterns in the Agile Canon

Daniel R Greening
Senex Rex
dan@senexrex.com

Abstract

Agility is the capacity of an actor (organization, team, individual) to rapidly sense, adapt and create to achieve goals in a rapidly changing economic environment. When an actor follows the first three base patterns in the Agile Canon—*Measure Economic Progress*, *Proactively Experiment to Improve* and *Limit Work-in-process*—it is agile. Its pace can keep up with change. Adding a fourth—*Embrace Collective Responsibility*—provides resiliency. The fifth—*Solve Systemic Problems*—expands agility beyond the actor boundary. To assess an actor, measure its conformance to these Agile Base Patterns. Agile researchers often frame their activities against the descriptive Agile Manifesto for Software Development. This has overly constrained their impact. The prescriptive Agile Canon pattern language is an alternative that could broaden and strengthen agile understanding and application.

0. Introduction

People often ask: *How agile are we?* The Agile Manifesto, the most recognized definition, covers only software, but doesn't really answer this question.

In this vacuum, breathless advocates contradict themselves. Agile is anything from no-rules self-organization to command-and-control [2].

But accepted agile methodologies have a goal. Agile methodologies seek to rapidly *sense* environmental change, *adapt* capabilities, and *create* solutions. I assert agile methodologies adapt to chaos [16] to gain competitive advantage. We grudgingly pay for agility because it helps us win.

Most creative activities can benefit from agility. As agilists make software production agile, we discover non-agile actors outside software impede our agility. But when most people encounter the term "software" in the *Agile Manifesto for Software Development*, they conclude "agile is a software thing." Previously proposed agile patterns have emphasized software [15].

Agilists study and teach many related methodologies—Scrum, XP, Kanban, Lean Startup, Getting Things Done, Pomodoro and Lean Manufacturing—because they reinforce each other.

I partitioned practices in these agile methodologies by how they produced agility. Five groups emerged. I then wrote an "agile base pattern" for each group.

I claim actors that *Measure Economic Progress*, *Proactively Experiment to Improve* and *Limit Work-in-process* are agile, but can find it difficult to sustain. Actors that *Embrace Collective Responsibility* gain resiliency. Actors that *Solve Systemic Problems* expand agility beyond their boundaries.

0.1. Pattern Languages

People struggle to learn large ecosystems of concepts, such as agile management. Concept descriptions in isolation make little sense, especially if they only teach us *How* to do something, not *Why* it works. Faced with this challenge, architecture professor Christopher Alexander developed a *pattern language* approach [3] to explain how to create healthy towns, buildings and construction [4]. We adopt his approach for agile practices.

Each pattern stands independently, describing a context, a problem, forces that constrain solutions, a solution that reconciles the forces and mitigates the problem, examples that use the solution and a resulting context. Patterns explain *why*; when we understand *why* a solution is needed, we remember and use it.

Pattern languages are collections of patterns organized from general to specific. They describe an ecosystem generatively. Reading just the first pattern of a pattern language provides value. Ideally, you can invent later patterns by applying earlier patterns.

The Agile Canon proposes a "theory of everything" pattern language for agile methodologies. The Agile Base Patterns described here are the generative beginning of the Canon.

With this effort we address the most obvious and general agile questions: Why do all these agile methodologies *work*? When do specific agile techniques apply? Is a specific effort, enterprise or person *agile*? We hope a general framework will help people answer field and methodology specific questions. We also hope to guide people to the right methodology for their particular situation.

Without further ado, let's get to the base patterns.

1530-1605/16 $31.00 © 2016 IEEE
DOI 10.1109/HICSS.2016.664

1. Pattern: Measure Economic Progress

We can study others who succeed, imitate their activities and gain their skills. But once we reach their capabilities, how can we improve further?

1.1. Plans don't guarantee creative success …

Creative efforts operate in an *economy*: a system where people manage limited resources to maximize return and growth. Economies drive everything. They need not involve currency. We can measure philanthropic efforts by the number of lives saved per unit of volunteer effort. We can measure companies by price-earnings ratio, market share or employee welfare.

1.2. Forces

1.2.1. Economies

Economic actors who operate without a well-defined economy wander aimlessly. They don't know what they value. They don't know their costs.

Whether individual, company or nation, economic actors without a compass will likely dither and fail as their unguided efforts cancel each other out.

1.2.2. Measurement

The most accurate measurements of success can significantly lag the completion of creative work. Lagging metrics applied to current decisions can fail perversely. For example, creative effort to build a strong asset can cost money. If this cost is characterized as a current loss, the company may invest badly [21].

Progress metrics (or "leading indicators") provide short-term guidance. We can use progress metrics to forecast short-term results, but predictions in a chaotic system worsen exponentially with time, making long-term forecasting impossible. Chaos even changes the utility of progress metrics: those that work now may perform badly later.

Progress metrics that are blind to significant risk, such as measuring untested and unreleased software production, often do not correlate well with desired outcomes, and thus produce bad decisions.

A single metric rarely serves to guide wise decisions.

Costly metrics inhibit frequent monitoring and produce inaccuracies. For example, detailed surveys with dozens of questions will skew measurements toward respondents with spare time.

Managers demand forecasts to make decisions, and many such decisions cannot be deferred. They prefer unqualified commitments, but chaotic economies produce probability distributions.

Considering too many metrics creates confusion and misalignment. Many organizations measure and report everything thought to be interesting, all the way up to executives. But this increases cognitive load and decreases decision-making quality.

Cognitive biases skew subjective measurements. They affect team estimates, customer surveys, employee performance assessments and more.

Every economy evolves: this year's resource limits are next year's surplus. For example, startup companies initially must show market traction. Later, these same companies must show profitability.

Variation accompanies creativity and chaotic economies. Aggressive attempts to control variation can destroy creativity. Well-intentioned process gurus applying Six Sigma to product development degraded innovation in several companies [13], including famous innovator 3M [25].

1.2.3. Rewards

When measurement drives reward, people game the measurements. People can game perceived value by simply omitting measurements.

Monetary reward fails to produce better creativity. Daniel Pink popularized the discovery that creativity improves when we reward people with mastery, autonomy and purpose [48]. This does not require us to discard economic metrics; it reveals that mastery, autonomy and purpose are major elements in our economy.

1.3. … therefore, measure economic progress with well-chosen, evolving metrics.

To "measure economic progress," articulate what you truly value, find metrics to detect progress or retreat, and measure frequently. The best economic progress metrics for agile actors are low-cost (so you can run them frequently) and fast (so you can adapt rapidly). Net Promoter Score surveys can measure loyalty in a single question [46].

1.3.1. Identify desired outcomes

Identify the biggest economic actor you influence (e.g., the company if you are an executive), examine its economy and articulate its goals. A mission statement lists timeless objectives, a vision statement lists multi-year objectives, and goals are short-term objectives (a year or a quarter). Work to reconcile them. Construct a concise, specific description of your current strategic activities at each time scale. Gain consensus from colleagues that everyone will seek these outcomes. The

better you can align desired outcomes, the more stable your metrics will be over time and the more productive your efforts.

1.3.2. Identify relevant metrics

Consider our goals for better measurements [37]. We seek better progress **measurements, which produce** better **insights**, which produce better **decisions**, which produce better **outcomes.**

Our mission, vision and goals describe desired *outcomes*. We work backward from those outcomes to identify progress metrics; hence this technique is called *ODIM* (outcome, decisions, insights, measurement).

Create a *metric suite*, considering these factors:

- Keep the number of metrics low. People cannot easily consider more than seven [44].
- Ensure measurements can be performed frequently without adverse organizational impact.
- Balance metrics to reduce perversity.
- Identify subjective metrics, and reduce bias.
- Consider growth rate, variance and other derivatives that may improve decision making.

Align goals from top (executive) to bottom (individuals). Start with the company's perspective first, then derive the department, team and employee desired outcomes and measurements. This harmonizes everyone's work [1].

1.3.3. Create a forecasting discipline

Establish an approach for forecasts. Scrum includes a bias resistant forecasting technique for delivery time based on velocity, story point estimation and "planning poker." However, it cannot responsibly model deferred risk. Monte Carlo simulation can improve forecasting results significantly [39].

Never report forecasts without including probability [38], especially when these forecasts are used for significant business decisions.

1.3.4. Embrace objectivity

Resist the temptation to reward personal creativity with money, as this will degrade metrics. Better motivators are mastery, autonomy and purpose. Even conservative oil companies have used these principles to guide performance management [33].

Encourage objectivity and learn from both success and failure. When failure has no cost, the greatest learning occurs when we challenge ourselves to achieve a 50% failure rate [51]. Reducing the cost of failure (such as by limiting work-in-process) can improve reporting. Be wary of "missing information" and encourage the organization to demand information from failures and successes alike. Executives can model the behavior they want by comfortably discussing their own failures, before asking employees to reveal theirs.

1.3.5. Evolve

Commit to a regular cadence to review and evolve metrics to meet emerging needs. Metrics that serve well at the beginning of a long project may not work toward the end.

1.4. Resulting Context

Actors using this pattern obtain coherent mission, vision and goals. They will use a handful of evolving metrics to gauge progress. They will learn faster from both failure and success. They will more accurately forecast future events by incorporating probability distributions. In short, they will better understand their economy and achievements.

2. Pattern: Proactively Experiment

Plenty of data informs us. Externalities affect our progress: competitors emerge, partners help us and delays harm us. We can forecast near-term futures.

2.1. We may not improve fast enough …

We suspect long-term dangers, economic loss, and growing ineffectiveness. Our friends reassure us, choosing their words carefully. Data is eerily stable. We aren't learning anything new.

2.2. Forces

2.2.1. Complacency Lulls Us

When we observe passively, allowing ourselves to be buffeted about by circumstance, we may wait a long time to discover new approaches. Competition and disruption motivate us to learn and improve, but when nothing threatens us we enjoy our lead and relax.

During periods of dominance, we often devote more effort to profitable operational (non-creative) work. We decreasingly worry about competitive threats. History is rife with complacent dinosaurs—Kodak, General Motors, Yahoo—who captured a lead, enjoyed a long dominant period, and then disruptive competitors stole their markets.

We may measure progress less frequently during stable periods, than during unstable periods. By the time we notice a problem we could be in trouble.

Operational work, the predictable stuff we can put on a checklist, is easy to do and measure. But it can consume all our time, if we don't limit it, leaving no time left for innovation and creativity. Manual tasks,

customer calls, goalless meetings all interrupt creative focus [6].

2.2.2. Loss of Control Scares Us

Failure risk accompanies innovation. Many people join dominant organizations because they value stability. They have learned to avoid situations with significant failure risk, and some overreact by avoiding all risky situations, even when failure cost is low. They prefer to allow others to take risks, and if disasters arise they can avoid blame.

Manufacturing gurus have told us that controlling variance leads to higher quality. But their definition of quality is predictability; when we try something new, it creates variance. When we follow their advice, innovation becomes harder and eventually stops [7].

2.2.3. Non-Creative Work Is Easier

Non-creative activities can easily be measured. Others have done it before, so we just compare ourselves to them. Here's a common metric people use: "How hard does it look like I'm working?" Unfortunately, for creative work, this can be a perverse metric. When we are working hard at non-creative work, we make noise, send emails, schedule or attend meetings, sweat, look worried. But maintaining these fictions defocuses us and degrades our creativity and others'.

For creative activities, just finding decent metrics can be hard. We want to try new ideas, and we learn the most when about half our hypotheses fail. Should we measure our failure rate, rewarding ourselves for a failure rate close to 50%? Or should it be lower, because it costs more when we fail?

Procrastinators tend to defer cognitive tasks, such as reviewing performance data and making decisions, with consequent poor performance and higher stress [27]. Strangely, procrastinators often perform better at cognitive tasks where there is no economic reward—it turns a chore into a game.

2.2.4. Uncertainty Confuses Us

Stephen Bungay argues three gaps interfere with planning in chaotic economies. A knowledge gap separates what we would like to know from what we actually know. An alignment gap separates what we want people to do and what they actually do. An effects gap separates what we expect our actions to achieve and what they actually achieve. These gaps mean that long-range plans are not likely to succeed without adaptation [12].

Despite these obvious planning uncertainties, we often think making more detailed plans produces better results. But over-planning consumes valuable effort and time, particularly when people demand details. Fear causes us to cling to the plan when unexpected things happen. And a detailed plan with confident dates and costs falsely reassures us.

2.3. … therefore, proactively experiment to improve.

If our environment isn't challenging us to innovate effectively, we have to challenge ourselves. Our economic metrics can help inspire innovation. What new things can we do to help more people, more effectively, and, get more value from our interactions? How can we work in new ways to produce more rapidly, cause fewer problems, and enjoy our work and collaborations more?

We can challenge ourselves by running adaptive "improvement experiments." General scientists examine historical data, brainstorm, choose a hypothesis to test, run a controlled test to validate the hypothesis, compare resulting data with hypothesized outcomes and draw conclusions. Improvement scientists do all those things, but focus on process changes, creative activities and prioritization to accelerate economic improvement. These improvement experiments test work hypotheses (such as "our lawyer can reliably deliver reviewed contracts within a week"), process theories (such as "pairing helps reduce rework and increase skill elasticity") and market theories (such as "millennials respond well to ironic ads") to create theories (hypotheses tested by experiments). In Scrum, for example, such experiments are constructed in the Retrospective Meeting.

Before changing the goals and processes of creative work, brainstorm different options and explore the possible results in your economic model. How will the changes improve your existing economic progress measurements? If you don't think they will affect progress measurements, but think you should make the changes, maybe you are missing an important progress metric to show the value of the changes. Consider revising your mission and economic progress metrics.

Experiments can be evolutionary or revolutionary [1]. Evolutionary experiments examine low-risk changes, and may validate the hypothesis after a short test. Revolutionary experiments examine more radical changes, and benefit from risk-mitigation. In the history of science, most experiments are evolutionary, punctuated with revolutionary upheaval when evolution stops working [35].

Once we establish a hypothesis, we construct an experiment plan. To keep our adaptation rate high, we should test our hypothesis with short iterative experiments. Evolutionary changes can be responsibly tested with just one or two iterations, while

revolutionary changes could require more. Even with revolutionary change, we should try to find results in the earliest iterations to help decide whether to proceed, change or abandon later experiments.

In teams, process adaptation helps balance perfect communication (expensive and slow) with unrestricted parallel activity (prone to mistakes caused by lack of coordination). I studied this phenomenon in parallel artificial intelligence applications [20]. Sometimes high communication generates better results, sometimes low communication generates better results. We can't identify those situations in advance, so we need to inspect and adapt.

Innovation causes variability. Low-risk variation can result in useful learning. When the economic metrics do not include metrics related to learning, the economic actor is likely to control itself into stasis, adapting but not innovating as conditions change. Ultimately, this makes the actor fragile.

Kaizen emphasizes small improvements. In a sense, PDSA and Kaizen are *greedy algorithms*, they always accept process changes that result in metric improvement and reject others. In chaotic systems, like markets and product development, greedy algorithms can result in locally optimal results but drastically bad global results. This explains why we should not rule-out revolutionary experiments, and expect some variation even in evolutionary experiments.

Variation always accompanies chaos and complex adaptive systems. Variation control is perhaps a great economic goal for mobile phone services (low-creativity), but not for mobile phone design (high-creativity). If we want to innovate in a chaotic system, our metrics must allow for and celebrate variation proportional to the chaos in the economy.

There are two solutions to these problems in chaotic systems. First, we must compensate for production metric variation by including learning metrics, and compensate for cost metric variation by including risk reduction metrics. Second, we must innovate at multiple scales, not just experiment with the small-scale changes advocated by Kaizen.

2.4. Resulting Context

Teams that rigorously apply experimental techniques can become *hyperproductive*, increasing their productivity and quality by factors of 4 or more [28] [29]. This could seem miraculous, but when observing teams, you'll find creative people spend substantial time communicating and learning (as well as blowing off steam). Many people use inefficient work techniques, but traditional operational management has no reasonable way to monitor and improve this, and can destroy creativity. When we shine an experimental lens on our creative, collaborative work, for the first time, we can improve much faster.

3. Pattern: Limit Work-in-process

We measure our economic progress and experiment with processes and products. However, experiments can take a long time, and failures can have huge costs.

3.1. We are going too slowly. Creating more detailed plans make things worse.

Our stakeholders, as a whole, make unsatisfiable demands. If we do not limit the number of things we start, we end up with many unfinished items and will, of necessity, fail more stakeholders and squander more resources than if we focused resources on the most important things and ignored less important things.

3.2. Forces

3.2.1. Inventory

Fungible assets can help us weather periods of scarcity, but increasing assets decreases cash flow.

Many people overvalue assets. Many creative assets are not fungible. Once they're completed, they earn a return on investment; before then, they are non-earning assets, which responsible people worry about. If we are operating in a highly chaotic market or production economy, these assets may hold us back.

In their favor, non-fungible assets can pose a barrier to entry against competitors, but only if the assets are at least as hard for competitors to overcome. This is rare.

People who fear failure often resort to more detailed planning [12], a costly, non-fungible asset. Detailed plans take time to develop, they are situation-specific, and until their offspring become viable, they have no tangible value. A plan that demands enormous investment before we validate its assumptions creates enormous risk [14].

Most people have a cognitive bias called the *sunk cost fallacy*: after we invest time or money in a project, we tend to continue investing, even if we discover that we won't get a decent return from the follow-on investment.

The combination of detailed plans and sunk cost fallacy produces terrible results in chaotic economies. We invest work creating a detailed plan, start executing the plan and then discover that slavish adherence to the plan will likely generate a loss. Because we have invested so much time, we continue following the plan, which could end in huge capital waste.

3.2.2. Congestion

When we saturate a roadway, our production line or our personal calendars, things slow to a crawl. Here's a quick test to see whether your company is agile: look at executive calendars. If they are full of meetings, the company is not agile. No one can rapidly sense the environment, adapt and create new things when their calendars are full.

As randomly timed requests to a system increase its utilization, the delay before a request is started (called "latency") increases exponentially. The delay is caused by requests being queued up behind earlier requests.

3.2.3. Cognition

Our most limited resources, as creative people, are time and attention.

People can remember a maximum of about 7 ordered concepts in one session [44]; more than this and people start making mistakes.

Many environmental factors interfere with cognition. Late in the day, after making many decisions or when blood glucose drops, people experience reduced capabilities, limiting their ability to choose [30]. Insufficient sleep degrades creativity and innovation [31]. Interruptions degrade creativity and quality [18].

3.3. ... therefore, limit work-in-process to improve value flow.

We take a broad view that work-in-process (WIP) refers to activities by large factories, by individual teams, by a person physically and by a person mentally. So work-in-process can refer to inventory in a warehouse or the number of things we are thinking about.

3.3.1. Cognition and Backlogs

A clear mind helps better prioritize work. Start by limiting your own mental work-in-process. David Allen argues by ridding ourselves of mental clutter we can reduce stress and improve creativity [5]. Nagging worries that we will forget to complete tasks and pursue opportunities clutter our minds and prevent us from focusing on high priority work.

By focusing on the most profitable work, we can earn new resources that help us with less profitable work. Scrum teams order a list of work items, called a Product Backlog, by relative return on investment (or profit). The most profitable items for the team's current situation appear at the top. Inter-item dependencies often create situations where it is more profitable to the team to finish a dependency before the thing that depends on it. But other times it's more profitable to complete a superficial version of an item, that doesn't require the dependency, because we can release the superficial version to discover whether customers will buy the sophisticated version [11] [50].

A discipline called Inbox Zero applies this Product Backlog idea to email [41]. It asserts that email inboxes are not ordered by personal profitability, and their disorder takes our valuable time and attention. Therefore, we must filter and move email items from the disordered inbox to an ordered backlog elsewhere.

Scrum teams limit work-in-process by creating a Sprint Backlog for work the team will focus on in a Sprint (typically a fixed time period of four weeks or less). The Sprint Backlog, like the Product Backlog, is roughly ordered by profitability, so the team attempts to work on the highest item first.

Our cognitive limits argue for people considering at most seven items at a time. In my experience, highly effective Scrum teams have Sprint Backlogs with about seven items.

Cognitive limits also suggest creating fractally structured Product Backlogs to help us rapidly see how projects could unfold. About 7 small Backlog Items appear at the top, followed by about 7 bigger ones, etc. Fractal Product Backlogs limit the amount of planning effort invested early in a long project, thus radically reducing detailed planning (a dangerous form of waste), decreasing sunk cost bias and encouraging rapid adaptation to new information about markets and production [22].

3.3.2. Collaborative Focus

Scrum and XP teams *swarm* on the topmost items they are working on. The goal is completion, and team members are discouraged from starting something unlikely to be completed by the end of the Sprint.

Scrum and XP require teams to produce a shippable product increment at the end of every Sprint. This limits production work-in-process and helps identify and limit technical risk.

Communication delayed can be a form of work-in-process. Scrum mandates a short daily Standup meeting to limit delayed communication between team members.

3.3.3. Value Stream Optimization

Kanban boards track active work by category and explicitly limit work-in-process for each category. The organization can then thoughtfully observe how work-in-process degrades work flow.

Organizations using Value Stream Mapping, which maps the time "from concept to cash" called *lead time* [49], can motivate work-in-process limits. Managers can identify the constraints where excessive

work-in-process is creating queuing or congestion delays, and limit work-in-process upstream to eliminate queuing and congestion at the constraint. These limits can reduce lead time all on their own, by reducing congestion, switching cost and queuing.

But now we can also increase flow by repurposing freed resources. If we limit work-in-process from the constraint through all upstream sources (called "subordinating activities to the constraint"), it helps us easily find available resources that can help increase the capacity of the constrained activity and increase flow rate through the system. This approach is called Theory of Constraints [19].

3.3.4. Incremental Work

To limit work-in-process, we can break a project into pieces and work on those pieces successively, but each completed piece must have value to an external stakeholder. This is called *incremental development*. If we modularize a project, and do each module in succession, we usually cannot deliver value to a customer until all the modules are done; this fails to limit work-in-process.

In large projects, managers often organize teams to focus on different modules, but this also fails to limit work-in-process. To propagate a change from a deep infrastructure module to an external stakeholder, elements must be handed from one team to another, with a queuing delay.

Feature teams compress parts of a dependency tree into parallel efforts, and thus reduce lead time [55].

3.4. Resulting Context

By limiting work-in-process, we can more rapidly deliver a single increment of new work to an external stakeholder, and get feedback. This helps us learn what the stakeholder will pay for, and adjust our plans incrementally to better meet stakeholder needs. When the stakeholder decides we've done enough, we can redirect our attention to another profitable project. Each new work increment also helps us better understand our production realities, and adapt to those. If we limit work-in-process sufficiently, we can profitably adapt to chaotic market and production economies; in short, we are agile.

4. Pattern: Collective Responsibility

It takes us time to decide to fix problems, and we let some problems fester because we don't want to get near them. When we are on a team, we can blame someone or something else for a problem, and often do. We might blame our own permanent flaws, feeling guilty. None of this blaming seems to fix anything, but we stick to our comfort zone.

4.1. We delay improvement by avoiding responsibility …

We are responsible for an outcome if our actions or inaction affected it. People assert we are "responsible for a failure," if we could have prevented it. We assert we were "not responsible for a failure," if we were not authorized or equipped to prevent the problem that caused the failure. People characterize us as "a responsible person" if we act to prevent or recover from failures.

4.2. Forces

We readily claim responsibility for success, but refrain from claiming responsibility for failure. It feels wrong to take responsibility for failures where we feel others had more control, and unfair to be held responsible when our passivity allowed failures to occur.

We like to operate in our *comfort zone*, where we can exercise our skills with mastery. This helps a collective effort when specialized skills are needed, but otherwise our specialties may contribute little to an outcome. People sometimes protect their special status by working in isolation, by hoarding specialist tasks, or by denigrating those with similar skills.

When demand for a particular specialty is high, a group may not have the capacity to meet the demand, and the whole effort may fail. When the demand for a specialty is low, a specialist could continue to produce specialized assets that aren't needed, creating waste and failure risks.

Members having greater agency—i.e., the capacity to affect the outcome—more readily take responsibility [52]. When others perceive us controlling a situation, it motivates us to act rapidly to fix problems and prevent future failures [10] [56].

When someone tells others they are inherently "responsible" or "irresponsible", it demotivates their future problem solving activities [32].

The Responsibility Process® claims that when failures occur, people progress through a series of unproductive stages before acting responsibly to solve problems [9]. Unfortunately, they can get stuck in any unproductive stage, leaving unaddressed problems to recur.

1. We may deny the problem.
2. We blame others.
3. We justify our actions by blaming circumstances.
4. We blame ourselves and feel ashamed.
5. We may feel obligation to keep "doing our job".

6. We may quit.
7. We take responsibility, and fix the problem.

Every state but the last is lazy. We do nothing to improve the situation until we reach the final state of *responsibility*.

4.3. … therefore, help people embrace collective responsibility.

4.3.1. Collective Responsibility

Collective responsibility is the notion that if any individual in a group can affect the group's results, we can attribute the successes and failures of the group to every individual. This sense of collective responsibility improves outcomes in many situations [26] [36].

Three conditions—autonomy, understanding and agency—allow us to assert a member has collective responsibility [34].

We can motivate members to act on behalf of the group, by better aligning their values with the group's values [47]. Avery argues that this alignment, particularly with small teams, is two-way: members bring their personal interests and values, and the group brings its goals and mission. When both member values and group values are negotiable, we can drive higher alignment [8].

We can increase agency by providing group members with broader authority, greater information or more training.

The *Responsibility Process* teaches us how to move more rapidly through irresponsible stages to reach collective responsibility. Being able to recognize the stage we occupy t tells us what to do next.

4.3.2. Culture

Organizational culture largely determines whether teams and individuals embrace and sustain collective responsibility. Most good leaders learned from (often spectacular) failures. Good leaders share those experiences to encourage followers to learn from failure rather than hide it.

4.4. Resulting Context

When team members operate from responsibility most of the time, everyone will likely complete more useful work, and waste less effort. Chronic dysfunctions will likely get resolved faster.

Collective responsibility motivates the development of broadly skilled colleagues. Broader skills create greater elasticity for uneven demand and avoid creating specialist assets before they are needed.

5. Pattern: Solve Systemic Problems

We measure, proactively experiment, limit our work-in-process, and embrace collective responsibility. When we operate independently of others, we can meet economic challenges and succeed.

5.1. External factors prevent us from adapting rapidly enough …

We don't have the knowledge, specialty resources, elasticity or authorization to do everything ourselves, but relying on others puts us at risk.

5.2. Forces

When we operate in a system with many actors, the dysfunctions of other actors can limit system agility, despite our best individual efforts. Myopia may focus criticism on innocents and leave causes unmitigated. For example, we often blame recently contributing actors for a late delivery, when common causes include executives failing to adjust the delivery forecast after a late start [40] and other teams producing poor quality components long ago.

We may compete for attention from dependencies, creating queues that dramatically increase average latency and, just as bad, increase variability. Even when a dependency can start immediately, few can accurately forecast progress. Virtually all dependencies have a risk that they will never complete (by the Turing incompleteness theorem), and this hidden danger can damage or destroy our efforts.

5.3. … therefore, collaboratively analyze and mitigate systemic dysfunctions

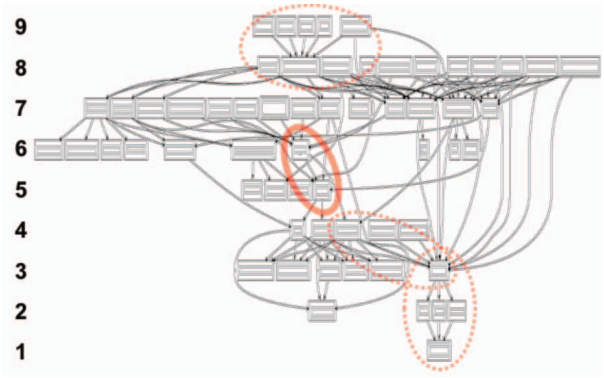
When we don't completely understand a system, we must engage others to help. When we don't have direct power over others, we must convince them to mitigate dysfunctions. We must collaborate.

5.3.1. Root cause mapping

If a problem involves many actors, bring stakeholders together to analyze it. Using the "five whys" method, work together in to deeply understand causes and identify the most effective fix [23].

5.3.2. Static analysis

Static dependency mapping helpa us understand and reorganize dependencies to reduce work-in-process, reduce lead time and increase quality [24].



Dependency mitigation candidates

For example, this figure shows the dependency graph of a large software system. Each oval shows teams that are candidates for merger. The solid oval encircles two teams that were ultimately combined, shaving 1 sprint off the lead time for changes made in levels 4 through 1.

5.3.3. Dynamic analysis

Dynamic techniques analyze the flow of activities through a system. Value stream mapping presents the value-add time, the non-value-add time and the queuing time for each actor in creating value "from concept to cash" [43]. We can then identify and mitigate systemic dysfunctions. Minor organizational changes to flow can often dramatically reduce lead time and improve other systemic progress metrics.

In the Theory of Constraints method [19] we leave the flow structure alone, but shift resources from the least constrained components to the most constrained. This approach advocates creatively retraining and repurposing people and equipment.

5.3.4. Teaching

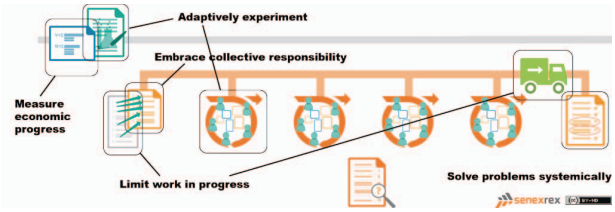
Seeking to fix external actors can be risky. We can't order them to change. We may try to explain their dysfunctions and implore them to improve. Sometimes this works; other times it threatens them. This can backfire to endanger agile programs and advocates.

Teaching agile patterns to external actors has many advantages. It helps them not only eliminate the dysfunction that concerns us now, but prevents others from arising later. Toyota famously taught its suppliers to use just-in-time manufacturing, because they were limiting Toyota's agility [45]. This built a infrastructure of just-in-time in Japan that accelerated its recovery from WWII.

5.4. Resulting Context

In applying this pattern, our ecosystem becomes more agile. When actors respond more rapidly and effectively, our own agility improves. We may interact more frequently with external actors. We may merge the activities of dependencies into our own work. We may find that long-standing problems disappear.

6. Scrum Example



Scrum events as Agile Base Patterns

This figure shows how the Scrum methodology implements the Agile Base Patterns. The first part of the Retrospective Meeting, shown at the upper left, assembles *economic progress metrics*, such as velocity, bug count and happiness [53]. The team tests their hypotheses: did they achieve the Sprint Goal and hypothesized velocity? Team members brainstorm process changes that might improve their metrics, hypothesize, and commit to the changes. Retrospective completion marks the start of Scrum's *proactive experiment to improve*. The Sprint Backlog *limits work-in-process* for the team. Team members *embrace collective responsibility* by swarming, pairing, and seeking help in the Daily Standup. During the Sprint, the ScrumMaster *mitigates systemic dysfunctions* (impediments), often outside the boundary of the team..

7. Summary

The Agile Base Patterns prescribe agile practices for any field or scale. Every agile methodology in this "big tent" analysis exhibits the five Agile Base Patterns. The first three Agile Base Patterns are necessary and sufficient for agility. The fourth adds agile resiliency. The fifth adds agile expansiveness.

The Agile Base Patterns arise from clustering roles, artifacts and events of agile methodologies, and are prescriptive. In contrast, the Agile Manifesto arises from the consensus of agile leaders, and is descriptive.

Colleagues using the Agile Base Patterns to assess organizations and guide transformations report favorably. I believe researchers and practitioners who incorporate these patterns in their work will improve their outcomes.

8. References

[1] Y. Akao, *Hoshin Kanri: Policy Deployment for Successful TQM* (2004).

[2] D.S. Alberts et al, *Understanding Command and Control*, CCRP, http://j.mp/1OAmxgQ (2006).

[3] C. Alexander, *The Timeless Way of Building* (1979).

[4] C. Alexander et al, *A Pattern Language: Towns, Buildings, Construction* (1977).

[5] D. Allen, *Getting Things Done: The Art of Stress-Free Productivity*, 2nd edition (2015).

[6] T.M. Amabile et al, "Creativity Under the Gun," *Harvard Business Review* (August 2002).

[7] R. Ashkenas, "It's Time to Rethink Continuous Improvement," *HBR blog* http://j.mp/hbrci (2012).

[8] C.M. Avery et al, *Teamwork is an Individual Skill* (2001).

[9] C. Avery by Amr Elssamadisy, "Christopher Avery on Responsibility," *InfoQ* (19 Nov 2009).

[10] A. Bandura, *Self-Efficacy: The Exercise of Control*, W.H. Freeman and Company, New York, 1998.

[11] S. Blank, *Four Steps to the Epiphany*. 2nd ed. (2003).

[12] S. Bungay, *The Art of Action: How Leaders Close the Gaps Between Plans, Actions and Results*, 2015.

[13] M. Cagan, "How to Kill Innovation," *Silicon Valley Product Group*, http://j.mp/killinno (Feb 2, 2009).

[14] R.E. Charette, "Why Software Fails," *IEEE Spectrum* (September 2005), pp. 42–49.

[15] J. Coplien et al, *Organizational Patterns for Agile Software Development* (2004).

[16] K. Dooley, "Complexity science models of organizational change," in Handbook of Organizational Change and Innovation, S. Poole et al eds (2004),

[17] M. Faggini et al, "The Failure of Economic Theory: Lessons from Chaos Theory," *Modern Economy, 2012:3*, 1-10, (January 2012).

[18] C.K. Foroughi et al, "Do Interruptions Affect Quality of Work?" *Human Factors* 56:7 pp 1262-1271 (Nov 2014).

[19] E.M. Goldratt et al, *The Goal: A Process of Ongoing Improvement*, 2nd revised edition (1992).

[20] D. Greening, *Simulated Annealing with Errors*, UCLA PhD Dissertation (1995).

[21] D.R. Greening, "Agile Capitalization," *InfoQ,* http://senexrex.com/agile-capitalization (2013).

[22] D.R. Greening, "Agile Pattern: Product Owner", http://senexrex.com/pattern-product-owner/ (2015).

[23] D.R. Greening, "Agile Pattern: Social Cause Mapping," http://senexrex.com/cause-mapping/ (2015).

[24] D.R. Greening, "Agile Enterprise Metrics," *Hawaii International Conference on Systems Science* (2015).

[25] R. Huang, "Six Sigma Killed Innovation in 3M," *ZDNet* (March 14, 2013).

[26] T. Isaacs et al, editors, *Accountability for Collective Wrongdoing*, Cambridge University Press (2011).

[27] E. Jaffe, "Why Wait? The Science Behind Procrastination," *APS Observer* http://j.mp/1Kk1EU4 (April 13, 2013).

[28] C.R. Jakobsen et al, "Scrum and CMMI – Going from Good to Great: Are you ready-ready to be done-done?" *Agile Conference 2009*, IEEE.

[29] C.R. Jakobsen, personal communication (May 4, 2015).

[30] D. Kahneman, *Thinking Fast and Slow* (2011).

[31] W.D.S. Killgore, "Effects of sleep deprivation on cognition," chapter 7 in *Progress in Brain Research*, Vol. 185, G. A. Kerkhof et al eds. (2010).

[32] A.N. Kluger et al, "The effects of feedback interventions on performance: a historical review, a meta-analysis, and a preliminary feedback intervention theory," *Psychological Bulletin* 1996;119 (2):254–84.

[33] C.G. Koch, *The Science of Success*, Wiley (2007).

[34] K. Kovach, "Genocide and the moral agency of ethnic groups," *Metaphilosophy*, Vol. 37, Nos. 3–4, July 2006, pp. 331–352.

[35] T.S. Kuhn, *The Structure of Scientific Revolutions*,1962.

[36] V.E. Lee et al, "Collective Responsibility for Learning and Its Effects on Gains in Achievement for Early Secondary School Students," *American Journal of Education* 104:2 (1996), pp. 103–147.

[37] L. Maccherone, "ODIM and Performance Index," http://j.mp/1NxqAcN (2012) .

[38] L. Maccherone et al, "Seven Deadly Sins of Measurement," http://j.mp/1KgfTZO (2014).

[39] T. Magennis, "The Economic Impact of Software Development Process Choice," *HICSS* (2015).

[40] T. Magennis, personal communication (2015).

[41] M. Mann, "Inbox Zero," Google talk, https://www.youtube.com/watch?v=z9UjeTMb3Yk (2007).

[42] K. Martin et al, *Value Stream Mapping* (2013).

[43] by Karen Martin (Author), Mike Osterling (Author)

[44] G.A. Miller, "The magical number seven, plus or minus two". *Psychological Review 63* (2): 81–97, http://psychclassics.yorku.ca/Miller/ (1956).

[45] T. Ohno, *Toyota production system: beyond large-scale production* (1988).

[46] Net Promoter Community, "The Net Promoter Score and System," http://j.mp/1QhVrJI (2015).

[47] L.E. Paarlberg et al, "Values Management," The American Review of Public Administration 37:4 (December 2007), pp. 387-408.

[48] D. Pink, *Drive: The surprising truth about what motivates us* (2011).

[49] M. Poppendieck et al, *Implementing Lean Software Development: From Concept to Cash* (2006).

[50] E. Reis, *The Lean Startup*, 2011.

[51] D. Reinertsen, "The Four Imposters: Success, Failure, Knowledge Creation and Learning," *Reinertsen Associates Blog*, http://j.mp/4imposters (Feb 1, 2013).

[52] M. Smiley, "Collective Responsibility", *The Stanford Encyclopedia of Philosophy (Fall 2011 Edition)*, Edward N. Zalta (ed.), http://j.mp/crstanford.

[53] J. Sutherland et al, *Scrum: The art of doing twice the work in half the time*, Crown Business (2014).

[54] Diane E. Strode et al, "A Taxonomy of Dependencies in Agile Software Development," 23rd Australasian Conference on Information Systems (3-5 Dec 2012).

[55] B. Vodde, "Choose Feature Teams over Component Teams for Agility," *InfoQ* http://j.mp/1gra3tW (2008).

[56] B. Weiner, Social Motivation, Justice, and the Moral Emotions: An Attributional Approach (2006).