Applying the Viable System Model to the challenge of Scaling Agile

Brief

As part of the drive to grow the Agile Business Consortium grows into a professional body it is important for it to build thought leadership in the field of business agility.

Specifically, there is a need to develop an approach to scaling Agile in organisations and the Viable System Model (VSM) has been identified as a foundation upon which an approach could be built.

As a result Patrick Hoverstadt, Mike Burrows and Steve Morlidge were asked to explore whether and how the Viable System Model (VSM) might be applied to this end. All three are experts on the VSM with experience of working with Agile methods. Their work is summarised in this and supporting documents and is aimed at decision makers and influencers within the ABC community, not a wider audience of Agile practitioners and leaders.

Note: in this document a capitalised 'Agile' (noun) is used to reference the set of well recognised practices that have grown up to facilitate the work of small teams, originally in software development. Non-capitalised 'agile', and 'agility' are used to refer to the quality of organisational responsiveness that may be related to but not confined to the use of such practices.

Background

The Agile movement has grown from its beginnings in the software development community into something that many organisations now aspire to be.

But it is naïve to believe that a whole organisation can become 'agile' by doing the same things at a different scale (as team of teams of teams etc) no matter how successful these practices have proven to be in a particular and much more specific context.

And organisations as a whole cannot become agile by nailing an Agile process or organisational unit to an infrastructure that is structured and run in a different way, based on some very different assumptions. Sticking a Ferrari engine in a bus doesn't make it a racing car.

It was with these issues in mind that we were asked to develop a framework to help organisations work out how to translate the principles that have proven to be so successful for developing and delivering software into a form that can be applied for all business activities at any organisational scale.

Fortunately, we did not have to start from scratch. The Viable Systems Model (VSM) provides a conceptual framework for scaling which can be developed to meet this need and we reconciled it with our practical experience of working with large, technology-dependent organisations. How VSM relates to the more general challenge of business agility is outlined in the companion document.

We will start with a statement of the problem we are trying to solve.

The Business Problem – the ultimate problem that needs to be solved

There are two main themes to the 'scaling up' challenge:

1. Agile methods have proven to be extremely successful at the level of delivery teams. However, it has proven difficult to maintain the qualities that make Agile successful at this level when the number of teams and the complexity of their work increases beyond a certain level.

2. Delivery organisations that apply Agile principles and values in their work often find themselves embedded in organisations with very different structures, work patterns, and behavioural norms. In practice, because of this unresolved disconnect it has proven to be difficult to manage the interface between the two domains and or spread Agile practices to other parts of the organisation.

The rise of frameworks such as SAFe, Less, Scrum@Scale and others shows that organisations are looking for answers to these challenges. Realistically though, no generic framework can be expected to be a perfect fit for every context, or to resolve every possible organisational tension. It would seem therefore that the means to understand and perhaps even manage the mapping between framework (branded or otherwise) and organisation should be of some value.

From a Viable Systems perspective, the underlying issue is one of complexity and relative rates of change.

Customer expectations change at the rate of the fastest supplier in the market so organisations are in race. And the weapon they reach for to help them out is Agile.

For large organisations, Agile has to be applied at scale, because just the odd team being faster is often not enough. The awkward end result can be Agile teams with a faster operational cadence than the departments they are part of, and departments that are locked into the typically slower cadence of traditional enterprises.

At the same time, at the level of teams, as the nature of customer demand and technical response becomes more complex the autonomy given to teams must become higher to cope, whereas the department and enterprise levels are to some extent insulated from this and the pressure to maintain organisational cohesion (alignment of activities) is stronger, creating a tension with the teams' need for autonomy.

Furthermore, different organisational levels are exposed to different parts of the environment, further exacerbating tensions that need to be managed. For example, the enterprise is very sensitive to the needs of financial stakeholders but is rarely directly exposed to consumers.

Together, the differentials in the rate of change, autonomy and perspective between the level of Agile teams and the rest of the organisation mean that doing Agile at scale creates huge tensions for any conventional organisation. Agile in a few teams is relatively easy to accommodate, but having an entire organisation working at different cadences, with different levels of autonomy and with different perspectives on the external world is much harder to deal with and in some circumstances can represent an existential risk to an organisation.

The fundamental challenge for any scaling approach then is how to reconcile those basic imbalances between organisational levels. Such imbalances can occur between units at the same level as well, but they are more common between levels.

One of the principles in VSM is the 'Viability Principle' which lays out the two biggest tensions for most organisations: the tension between autonomy and cohesion and the tension between stability and change. Where these tensions are not resolved well enough, organisations collapse. VSM sets down the systemic criteria for managing these tensions through time. In the case of scaling Agile, both these tensions can be stretched to breaking point with differentials in both change rates and autonomy between organisational levels. This is why this is such a big problem both for organisations seeking to scale Agile and for the Agile community as a whole.

The problem – what scaling frameworks do and don't provide

Scaling frameworks seek to provide well-defined processes, structures, and practices that allow Agile teams to coordinate together in a larger delivery organisation. Depending on the framework, the scaling applies horizontally, adding capacity inside the scope of an existing process, and/or vertically, adding layers of structure in one or more dimensions – organisational structures, work breakdown structures, and structures for activities such as planning and monitoring.

The challenge is to provide structures such that teams are able to coordinate and collaborate effectively where that’s most needed, and without incurring unacceptable costs and delays. Adding structure does not in itself increase agility, and without significant effort to mitigate its effects, the tendency is very much in the other direction. The more structured the organisation or process, the less opportunity there is for ad-hoc collaboration. The more structured, standardised, and more centralised the work breakdown structure, the harder it is to change. The more layers involved in decision making, the slower decisions are made. And so on.

Furthermore, these layers of structure leave teams increasingly separated from the strategies that they are required to implement. Indeed, the scaling frameworks typically regard the strategy process as outside of their scope. If strategy is something that happens to you, how much autonomy do you really have? And without some meaningful distribution of autonomy, how is any kind of agility – let alone any principled form of Agile – possible?

Our Process

In order to demonstrate and assess the usefulness of the VSM in this context we applied it to an organisational archetype of the sort that followers of the ABC will be familiar with.

The primary purpose of our hypothetical organisation is the development of software, or a service where software is a major part of the value delivered to the customer and to the company.

The fundamental organisational units are made up of Agile development teams. Their work is orchestrated at a portfolio level (‘Teams of Teams’), while another set of more generic corporate activities is managed at an Enterprise level.

The VSM describes what needs to be done to manage the activities within each of these levels. But, as discussed, managing the relationships between the levels is the key to successfully ‘scaling Agile’, so particular attention is paid to this aspect of the modelling process.

By doing this we hope to bring the model to life. Along the way we will explain how familiar Agile practices map against the VSM and how and why ‘local’ solutions are often developed to supplement them. At the same time, we will use the VSM as a diagnostic tool to analyse some of the typical shortcomings of scaling frameworks.

Introducing the Viable Systems Model

The Viable System Model (VSM) was developed in the 1970’s by Stafford Beer, drawing on biology, information theory, and cybernetics. It claims to be a universal model for systems that seek to maintain a sense of a separate identity and thus a drive to survive by adapting to its changing environment. And in the five decades since its inception it has been well tested in a wide variety of domains and at a wide range of scales.

The model does not prescribe a particular design. Nor is it a framework to implement. Rather, it’s a lens through which an organisation can be viewed. It is descriptive and diagnostic, identifying what must necessarily be happening in some form in a viable system, and helping to identify deficiencies, for which solutions can be developed.

Its diagnostic power and usefulness as an aid to design comes from:

- The six necessary *systems that it identifies* – dysfunction arising from the absence of any system or a deficiency in how they work
- The relationships between those systems – dysfunction arising from ineffective communication or from imbalances of quantity or authority
- The relationship between systems at different levels of organisation – for example between teams and teams-of-teams, or between department and division. Very often those relationships barely function, and where they do function, one level is trying to do the work of another above or below.

A consequence of VSM's universality is that it is more abstract than most models of organisation. Therefore, to make it more easily relatable, we need to consider what must necessarily be happening in an organisation that is deliberate about its adaptability and show how they are reflected in the VSM.

Specifically, if the organisation is big enough for scaling to be relevant, it should be possible to identify certain things happening at multiple levels of organisation. Up to the scale of the organisation as whole, for each of these levels:

1. There must be useful work being done in accordance with its current commitments. In order to perform that useful work there will be appropriate communication between the people or organisational units that comprise that level as it is currently structured. And in the process of that work, intelligence and insights will be gathered and shared.

2. There must be trust between those people or organisational units – trust for example that their respective internal commitments and activities reflect agreed strategy. This trust is built through mechanisms such as transparency, self-reflection, self-description, monitorability, auditability, and so on

3. Work commitments will be continuously adjusted through a process of learning and to align with the opportunities it has developed for itself and chosen to pursue.

It is not hard to see that the organisations, teams, and levels in between could be significantly compromised by problems with any of the above, though it should be noted that the correspondence between effective levels and the organisation's formal structure may not be obvious.

Between levels, it suggests that trust-building must extend deep and wide, and further asserts that there must be some negotiation between and across levels as strategies develop.

The Viable System Model further asserts that in a well-functioning organisation, activities will be structured to match the complexities of its external environment as well as reflecting its own internal strategies. Therefore, one cannot describe or prescribe a successful organisational model without taking its external environment into account.

The VSM’s six systems below are easily identified in the above three-part description[1]. The numbers are how Stafford Beer labelled them; for the sake of relatability we have named them also:

- **System 1 (Value Creating)**: Where the productive work gets done, conducted by multiple viable operational units, structured in an appropriate way, and each unit relating to the organisation’s external environment
- **System 2 (Coordinating)**: The tools, practices, and protocols that exist to maintain stable operation – synchronising activities as required, avoiding wasteful competition over scarce resources or (conversely) overburdening the organisation.
- **System 3 (Organising):** Managing performance, ensuring that resources and capabilities are well matched to agreed commitments, organising to create synergy.
- **System 3* (Monitoring)**: The typically sporadic practices that give the assurance that Systems 1-3 are together working as they should and thus help build trust.
- **System 4 (Adapting)**: Activities dedicated to surveying the external environment for opportunities and threats and to developing possible responses.
- **System 5 (Governing)**: Keeping value-creation and developmental activities in appropriate balance and ensuring that both are in accordance with the organisation's identity and purpose

The VSM asserts that, in the interests of viability, all six systems must be present and functioning effectively at all levels. Thus an organisation is made up of a set of nested, interlocking viable systems with the same regulatory structure, tuned to the specific needs of their respective environments.

One important last thing to note: being descriptive and diagnostic (as opposed to prescriptive), the model does not prescribe any relationship between systems and the formal roles that people may adopt. The model works for assessing completely self-organising adhocracies, for traditionally differentiated and managed forms of organisation, and for shades and hybrids in between.

How the VSM helps to ‘scale Agile’

The Viable System Model is a model of effective organisation that can be applied in any domain, based on sound system-theoretic principles. And it explicitly addresses how complexity needs to be managed to maintain any organisation’s viability – its ability to survive and thrive in environments that are, to varying degrees, dynamic and fundamentally unpredictable.

VSM is a mature model that has been extensively applied in a wide range of organisational settings. In our context it is worth drawing attention to several features that are complementary to Agile principles:

[1] Description based on the Deliberately Adaptive Organisation, deliberately-adaptive.org

- The ultimate source of organisational viability is deemed to be a healthy relationship with its environment (within which customers hold a special place in Agile).
- To manage complexity, it prescribes balancing local autonomy consistent with the need for organisational cohesion.
- It allows for effective organisation being achieved through means other than hierarchical, power-based relationships
- Instead, it shows how to structure and manage work at different scales using a recursive (fractal) structure. Because it has features that are self-similar at different scales it is relatively simple but has the sophistication required to model large, complex organisations.
- Nevertheless, it explicitly allows for different levels to be structured and managed differently as appropriate to their context. For example, it does not expect the specific routines, rhythms and practices of one level to be applied in the same way at other levels.

In summary, a scaling framework based on the VSM will take the form of a set of patterns and principles which would guide where, when, and how to apply tried and tested Agile methods and also to identify gaps that need to be filled. As a result, it can be used to describe, diagnose or as a guide to the design of effective organisations.

The results of applying the VSM to our organisational archetype are described in detail in the Technical Supplement to this document. The key findings and conclusions of our work are summarised below.

Findings

1. The mechanisms and process that have evolved to organise work done within team level Agile is highly consistent with the VSM.
2. Most scaling frameworks either ignore the Enterprise level of recursion or assume that the practices are fixed and non negotiable. This is despite the fact that typically Enterprise level practices are built on inflexible top down annual planning processes (based on the same ethos as waterfall planning) which are alien to Agile with its emphasis on customer needs, continuous iteration, learning and change and a high degree of decision making autonomy.
3. Since scaling frameworks do not address the need for or nature of changes required at Enterprise level to make the whole organisation more agile, they fall short of the promise to ‘scale Agile’.
4. As a result, typically we either see a fracture in between the Portfolio and Enterprise levels or conflict on the boundary between teams and the Portfolio level if they try to manage Agile teams applying ‘Enterprise’ norms. This typically manifests itself as one level trying inappropriately to manage the work of another, either up (second guessing) or down (micromanaging).
5. At the team of teams/portfolio level many of the mechanisms that the VSM identifies as essential to reconcile the existence of largely self managing teams with the need for organisational cohesion are either absent or built in an ad hoc way in response to local issues, rather than being consciously designed to meet a well defined need.

6. There is often a failure to recognise the need for or legitimacy of strategy and governance frameworks at all levels with the consequence that either Agile teams feel like strategy is something that is 'done to them' and local teams can develop a sense of identity that is antagonistic to the rest of the organisation.
7. There are a raft of approaches and processes already used in an ad hoc way in some Agile contexts that could be used to systemically address the shortfalls in existing scaling frameworks highlighted by VSM.

Conclusions (so what?)

Our main conclusions are fivefold:

1. The VSM helps understand common problems that organisations have with organising work at scale. It also helps to forensically diagnose specific issues and can be used to specify solutions.
2. It also exposes shortcomings with existing 'scaling frameworks'. Some of these deficiencies are real - for example the failure to deal with the interface between Agile and traditional corporate management practices. But they also reflect the difficulty of applying any sort of prescriptive process framework to diverse and complex set of organisational environments. In this regard a principles based approach such as that offered by the VSM will always work better.
3. Since the VSM stresses the importance and the dynamic nature of the relationship between an organisation and its environment at all levels, it is philosophically aligned with the Agile movement, and thus a good fit.
4. VSM is well suited as an approach for scaling Agile in organisations of any size or complexity,
5. There is sufficient evidence to commission additional work to develop a VSM based scaling 'product'.

Technical Supplement

This Technical Supplement sets out in detail how the Viable Systems Model can be used to understand, analyse and critique the working of an archetypal software based service company of the sort that will be familiar to members of the Agile Business Consortium.

The aim is to illustrate how the VSM can be used. Less technical language, more examples and explicit guidance will be required if and when this is turned into a 'product' that can be widely deployed.

RECURSIONS (Fractal Levels)

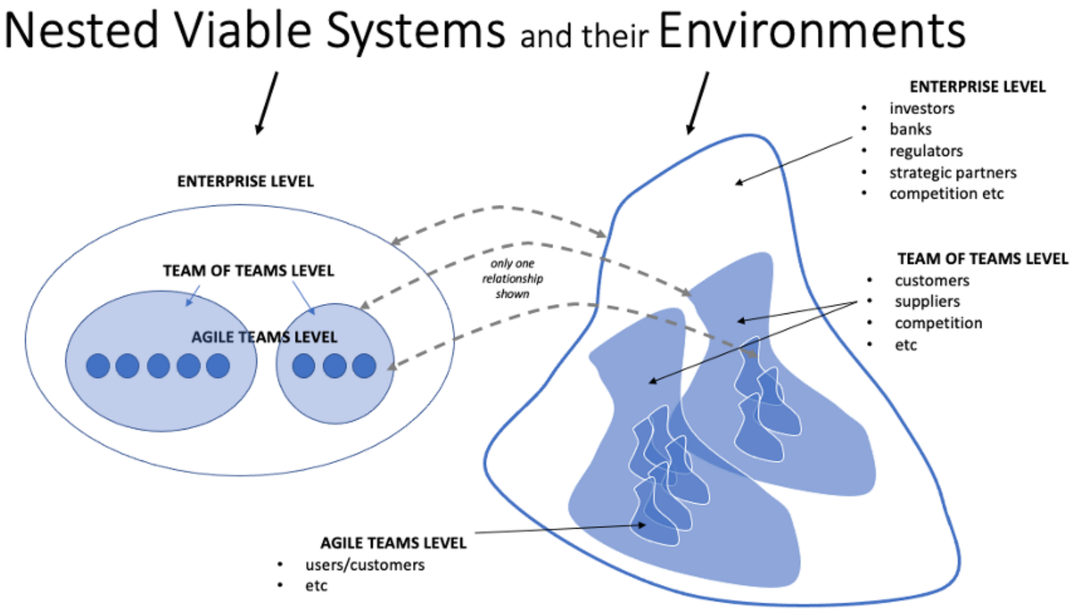
The first step in applying the VSM often involves identifying levels of recursion.

Small organisations may only have one level. But larger ones will have many, each with its own identity reflected in the differing nature (content, detail, cadence etc) of the work that is performed, the different set of external relationships and differing perspectives on the organisation and its future.

Any System 1 component (I.e., operational unit) at a given level has the potential to become independent[2].

To be effective, nesting is driven not by reporting lines but by the need to find a structure of business domains that for all concerned organises not just the work but the work's relationship to the external environment, a major source of complexity, as the chart below illustrates.

[2] For this reason, support units are not considered to be part of System 1. Because they do not have a set of external relationships based on value exchange and should not conceive of themselves as having an identity distinct from the organisation they serve they are not considered to be 'viable'.



The nesting of viable systems and the formal structure of the organisation often differ. People can participate at multiple levels, multiple levels of organisation may be required to all of fulfil the viable system's necessary activities, and the relationships between levels are mediated via multiple channels.

ILLUSTRATION: SOFTWARE BASED SERVICE COMPANY

In a large software company, we might expect to see at least three levels of recursion as shown above. How the relationship between these levels are managed will be addressed later, after all the elements of the Viable System have been described.

Each level has a distinct set of environmental relationships relevant to their differing roles. And just as there are relationships between different levels of organisational recursions, there are relationships between different environmental levels that the organisation needs to manage, where there are likely to be tensions, for the reasons already discussed.

In the case of a software company, it is usually not difficult to identify the recursions. At the lowest level, there are Agile Delivery Teams supported by a rich repertoire of methods. At the highest level is the Enterprise, and between them is the Team of Teams level, both of these usually easily identified and distinct from Agile teams.

Findings

There are often issues identifying appropriate methods for managing activity at the ‘Team of Teams’ level and managing the relationships between this level and those ‘below’ and ‘above’ (Agile Teams and the Enterprise respectively).

Getting the operational structure ‘right’ is the biggest single factor affecting organisational efficiency.

Note: all the diagrams in the body of this document show simplified and selective views of the Viable System Model. A comprehensive representation can be found in the Appendix.

QUESTIONS TO ASK:

- Does the organisation's operating structure reduce the dependencies between teams?
- Is the way teams are grouped together clear,
- does it:
  - support teams in doing their work
  - and allow a clear line of sight between the teams' work and the Enterprise's higher level objectives?

SYMPTOMS OF FAILURE

If different levels of recursions are defined incorrectly, we might expect to see:

- The work of one level being done at another. This might be manifest as bureaucracy, micromanagement, or second-guessing
- Inappropriate patterns of work, for example operational ‘flow’ being disrupted by financial periods.
- Decisions being taken without the requisite knowledge, e.g. determined by rank.
- Confusion over what work gets done where and where decisions get taken

SYSTEM 1 – VALUE CREATING

The fundamental building blocks of the VSM are the elements of System 1. These contain those activities that produce things of value to the outside world. All but the very smallest organisations have multiple value generating units in System 1.

The VSM requires that System 1 units at a given level be defined so that they are capable of being largely self-regulating. This means that:

- The boundaries between them should be drawn such that the interdependencies with other systems be minimised.
- They are likely to contain people with different skill sets.
- Their size will be constrained by their ability to self-regulate to a large degree.
- They will have considerable decision-making authority, matched with accountability for output, both of which will be clearly defined.

However, by definition, in any organisation there will be some dependencies between the elements of System 1. The VSM requires that these be managed in a way that imposes the minimum restrictions on their autonomy.

Some interdependencies can be managed directly by bilateral collaboration (e.g. if Unit B is supplied by Unit A), in which case the relevant units should agree appropriate arrangements. But most coordination tasks involve multiple units. This is the role of System 2.

System 1: Operational Units and their Relationships*

* not all shown

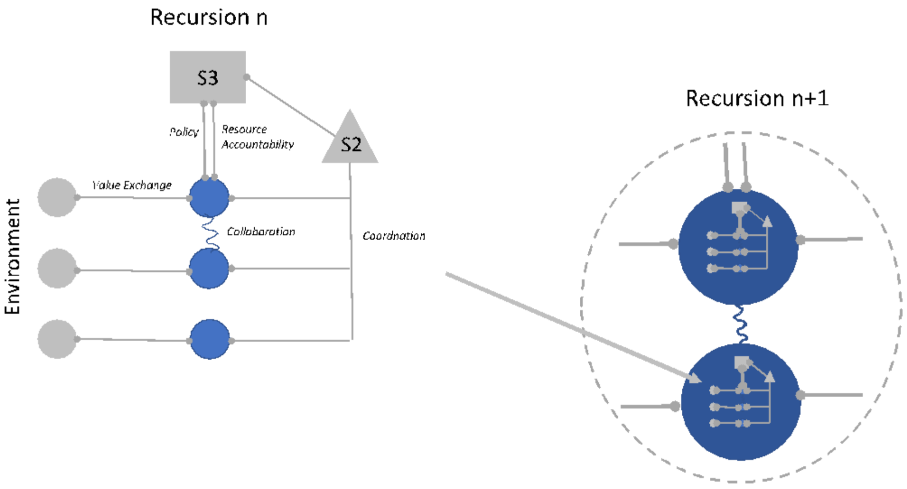


ILLUSTRATION: SOFTWARE BASED SERVICE COMPANY

Small operational teams, each with clearly defined areas of responsibility, is a foundational principle of Agile. At the level of Teams (Recursion X) they deliver value in the form of software functionality. At higher levels of recursion, the value delivered may be a piece of software or a service. And at the Enterprise level in a public company the value exchange is often mainly manifest as financial transactions (funding, dividends, capital appreciation and so on).

A key consideration is how operational units below the Enterprise level are defined. The way that the teams at the team or portfolio levels are defined could reflect:

- The nature or the focus of the work (e.g. a product, service or process)
- Maturity (start up, development, maintenance)
- Geography
- Customer segments

….or a mixture of them, depending on circumstances.

The hard and fast rule is that they should be defined such that they are capable of self-management to a large degree (i.e. with minimal interdependencies), consistent with the overall resource constraints.

This means, amongst other things, that the majority of the burden of managing the coordination of activity is borne by teams which should be sufficiently small that it can be handled quickly and with minimum effort, often informally, particularly if team members are co located.

Once a team at any level of recursion finds that it cannot manage organisational complexity effectively this can be a sign that the unit needs to be broken up or, another level of recursion added.

The fact that every operational unit has a relationship with the external environment where value is exchanged means that in principle any operational unit could exist independently. This distinguishes operational units in System 1 from those that support the organisation. But for as long as it is part of the organisation it needs a healthy set of relationships with other Viable System components in order to perform effectively.

FINDINGS

In an organisation employing Agile practices, System 1 teams are usually well defined at the level of delivery teams. But at other levels of recursion they are often poorly defined or misaligned with the value creation structures of an organisation.

Where teams have been formed with unnecessarily high levels of interdependence, inefficiency is high, adaptability is low. And in software companies for example, this can be the single biggest driver of bugs in software.

QUESTIONS TO ASK:

- What is the degree of interdependency within teams relative to the interdependency between teams?

- What is the degree of interdependency within units at the ‘team of teams’ level relative to the interdependency between those units?
- Is the grouping of teams and units aligned to the strategic needs of the enterprise?
- Is the grouping of teams and units aligned to the demands from the environment?

SYMPTOMS OF FAILURE

Internal conflict, excessive demands for coordination (e.g., in the form of meeting proliferation) or need for external intervention or mediation can all be signs of poorly defined System 1 operational units.

This will occur:

- If teams are too large
- If teams are too small
- If teams do not have the right balance of skills, perhaps because they are organised on functional lines
- If teams are unable to readily coordinate their internal activities
- If teams do not have well defined things for which they are responsible
- If teams do not have adequate decision making authority.
- If the way teams have been grouped doesn't match the task or the needs of customers.
- If there are too many interdependencies between teams.

SYSTEM 2 – COORDINATING

Interdependencies arise whenever the operational units of System 1 at a given level of recursion:

1. Draw on shared resources,
2. Need to collaborate to achieve a shared objective (internal or external) or
3. Are part of a shared process.
4. Share an environmental niche, e.g., serve the same customer or market.

Having identified the interdependencies, mechanisms need to be put in place to facilitate coordination so that the work of one unit does not disrupt, compromise or delay the work of others. The nature of the mechanism required will vary. For example:

- Where the need is **frequent and simple**, passive mechanisms can be used e.g., by defining common standards, timetables, definitions, protocols etc.
- Where the need is **complex but infrequent**, mechanisms for mutual agreement are required.
- Only where the need is **complex and frequent** are formal mechanisms required to actively manage coordination, e.g., regular formal meetings etc.

The role of System 2 is to provide the operational units (in System 1) with the ability to coordinate their actions while minimally constraining their ability to govern themselves.

Operational units can and do engage in coordinating their activities without the need for System 2 particularly when the dependencies are bilateral. And they may be forced to devise system wide informal System 2 mechanisms if the 'official' ones are deficient or absent.

Since System 3 has the legitimacy, perspective, and resources to define, set up and oversee the full set of System 2 activities, any failure in this regard lies at its door.

ILLUSTRATION: SOFTWARE BASED SERVICE COMPANY

In a software business that uses Agile methods there are many well defined ways to coordinate activities within a delivery team. These include:

- visual management techniques
- regular stand ups
- standard practices for ways of working and coding practice and language to ensure technical compatibility.
- strong team rituals that build team cohesion and awareness at a behavioural and work level.
- techniques to avoid overburdening, minimise context switching and to promote pairing and swarming behaviours.

And of course, co-location and the small size of teams are great enablers of informal coordination.

At the level above – the Team of Teams - establishing a set of appropriate and effective coordination mechanisms between delivery teams is the key to effective scaling of organisations.

The list of potential mechanisms is long, and could include

- continuous integration, interface design, standardising APIs and software architecture all of which reduce interfacing problems for both tech and process.
- synchronised team cadences reduce the scope for conflict and
- teams working on the same products or customers and Spotify 'guilds' reduce team divergence by setting standards.

A fuller set of examples is included in the appendix.

At the Enterprise level in a traditionally run business, coordination is often based on annually agreed activity plans supported by a set of fixed budgets. Since these are difficult to change more frequently than once a year and don't address activities that bridge financial years, such practices cannot be called 'agile' or 'Agile'!.

Alternative models, such as Beyond Budgeting advocate managing activities across a rolling horizon and at a cadence that is in tune with the frequency at which the environment changes.

FINDINGS

Coordination is 'baked in' to Agile practices at the level of delivery teams (eg daily stand-up meetings) and there are many Agile processes that serve this purpose at the 'Team of teams' level (eg portfolio kanban systems). However, in a traditional organisation the need for coordination at the Enterprise level is often not well understood and what mechanisms there are often do not sit well with Agile practices elsewhere in the organisation.

QUESTIONS TO ASK:

- What are the ways that one operational activity can adversely affect the work of another operational activity and are there effective ways to prevent that happening ?
- Are you using costly and intensive resources (e.g. meetings) to resolve commonly occurring low level inter-unit issues?
- Are you experiencing: turf wars / bottlenecks / miscommunication / handoff problems / quality issues / scheduling conflicts between units?

SYMPTOMS OF FAILURE

A failure to establish appropriate mechanisms of coordination is the most common failing identified in organisations. Symptoms include:

- frequent or recurrent escalation of coordination issues to 'management' (S3),

- excessive amounts of time spent in unproductive meetings
- turf wars
- a proliferation of ad hoc attempts by operational units to align the work of peer groups
- inter-team quality issues
- operating to different standards
- mis-communication
- poor process ‘handoffs’, long wait times and rework.
- reliance on planning or chasing things that could be managed by peers or through passive co-ordination methods
- over centralisation of coordination, where other levels/units intervene where matters could be self managed.

SYSTEM 3 – ORGANISING

In the VSM the role of System 3 is to ensure that operations and capabilities deliver the performance the organisation needs. Since resources are always limited, being granted access to them always involves the operational units being accountable for using them efficiently and effectively in line with interests of the organisation as a whole. This is usually manifested in goals or objectives.

The core role of System 3 is to manage these critical activities on behalf of the organisation in a way that strikes the right balance between:

- the local (System 1) autonomy and agility required to effectively respond to the demands of the environment and
- the need for organisational cohesion.

System 3 normally operates in the here and now or relatively short term.

It typically will make design decisions such as how best to structure the operational units of System 1, and establishing and maintaining System 2 and System 3*.

Also, System 3 is responsible for identifying and exploiting potential efficiencies and synergies. These could involve:

- Knowledge: exploiting useful knowledge generated in one place and spreading it.
- Shared services: where it is not economic for individual operational units to be self-sufficient (e.g. if an activity is not central to the role of a unit, and/or if economies of scale outweigh the benefits of localisation).
- Ensuring operational units cooperate where necessary to deliver performance.
- Regulating the flow of resources (people, money etc) to operational units to enhance the effectiveness of the organisation as a whole.

ILLUSTRATION: SOFTWARE BASED SERVICE COMPANY

in ‘traditional’ Agile at the level of Teams, resources and time are treated as fixed, and requirements are flexed.

Thus, resource management is confined to allocating individual team members to tasks which within a team can be accomplished relatively easily. Performance management is a combination of task allocation (‘working software to do X’) strong emphasis on self-inspection, team retrospectives, trust building and burndown rates, and process measures such as conformance to timeboxes etc.

System 3 practices consistent with Agile principles at the ‘Team of Teams’ level include:

- Structuring of S1 operations/teams to minimise the coordination load.
- Setting the performance cadence and horizon.
- Designing appropriate System 2 mechanisms to enable delivery teams to coordinate their actions without the need for System 3 involvement.
- Defining decision rights (with the goal of providing delivery teams with the authority they need to work in an ‘Agile’ manner)

- Defining technical standards and infrastructure such as code, data architecture, technical architecture.
- Providing teams with services where they cannot be self sufficient, e.g. shared support services (HR etc.) and specialist resources not available in teams.
- Allocating resources to teams, including people, money, hardware, and setting the performance criteria (which may be quantitative or qualitative).
- Monitoring performance against these criteria.
- Intervening when issues are identified (in performance or by Systems 3*) that cannot be remedied by delivery teams themselves.

In practice, however, activities at the 'Teams of Teams' level often more closely resemble those associated with traditional management at the Enterprise level, manifested as 'top down' planning and control and fixing resources on an annual cadence and horizon.[3]

The inability to reallocate resources in response to emerging threats and opportunities can become a brake on organisational agility. This is particularly problematic when there is a high degree of uncertainty in the environment or in the nature of the work, for example in managing innovation and 'startup' units.

The resource allocation role of S3 is further complicated by the tension between driving efficiency in delivering current services and investing resources to ensure capacity to deliver against future needs. There is almost always a tension between efficiency in the short term and effectiveness or relevance in the longer term. This tension is at the heart of the agility problem.

As well as managing the best distribution of resources, the second major tension S3 needs to deal with is the distribution of decision making and this is at the heart of the autonomy - cohesion debate.

If teams have too little autonomy then they don't have the flexibility to meet changing demand. If they have too much, then it becomes difficult to take strategic decisions or to act strategically when the organisation needs to. Operational and tactical decisions taken at the team level can simply override strategic decisions at the Team of Teams or Enterprise level if those levels have ceded all control to teams.

An additional challenge is that traditional management practices are at variance with the principles and culture associated with Agile, which can cause conflict and resentment.

FINDINGS

As discussed, the inherent conflict between traditional 'top down, inside out' plan driven management practice and the evolutionary 'bottom up, outside in' approach of Agile often manifests itself in the domain inhabited by System 3.

[3] This is what the Beyond Budgeting model seeks to remedy.

There are two ways to resolve this. You can either consciously design practices and processes to integrate or align the two stances at the interface between the two ‘worlds’, with the aim of minimising the damage.

Or you can ‘scale up’ by building the whole organisation around a model designed to balance the needs of both levels. The VSM provides a framework to guide this approach and models from related domains (e.g. Beyond Budgeting) can help. But trying to resolve this by simply copying and pasting approaches designed for teams to the level of the whole enterprise do not work because the scale and nature of the issues that need to be addressed are fundamentally different.

This is dealt with in more detail in a later section on cohesion.

QUESTIONS TO ASK:

- Is the balance of agency (ability to act) between levels proportionate to the challenges each faces - bearing in mind that those demands will fluctuate constantly?
- Do both teams and Team of teams levels have the decision rights that are suitable for their situation?
- Are there any decisions that are taken at a higher level that could be sensibly taken at a lower level?
- Does each level have the information it needs to take the decisions it needs to take?
- Are decision rights clear between levels with no overlaps and no gaps?
- Do teams get the resources they need and is there a clear way for them to negotiate the resources they need?
- Are any teams over-resourced relative to their performance?
- Is synergy being delivered?

SYMPTOMS OF FAILURE

Where there is a poorly managed interface between ‘top down’ traditional management practices and ‘bottom up’ Agile ones this can be manifest in resentment, suspicion and conflict. For example, attempts to impose conventional enterprise approaches such as waterfall business planning on Agile Teams or conversely trying to use approaches designed to work with small teams at the level of a large organisation will generally not work well.

The nature of the responsibilities exercised by System 3 means that it has a lot of power within organisations, and many of the problems we see flow from the misuse of the power, at least from an Agile perspective. These include:

- a tendency to micromanage, i.e., do the work of a lower level of recursion
- monopolising decision making rights
- excessive bureaucracy
- support functions acting in their own interests

- slow decision making
- decision conflicts (who should take decisions)

SYSTEM 3* - MONITORING

In principle, Systems 2 and 3 should be designed in such a way that all activities in the 'here and now' are managed in a balanced manner with minimal need for intervention.

But no system is perfect, things change and things break. The role of System 3* ('three star') is to identify flaws that System 3 would otherwise be blind to, thereby enabling it to deal with emerging issues before they create major problems.

To be effective in this role System 3* needs to

- Have access to see operational activities in their normal state.
- Be perceived as building mutual trust, supportive rather than intrusive or controlling.
- Work to maximise the chances of being exposed to issues that have not been anticipated in the design of the system.

Done well this process builds trust. Managers will properly understand operations and know that they can trust what they are told. And teams know that management is interested in knowing what is going on and how things really work.

But done badly, it can destroy trust.

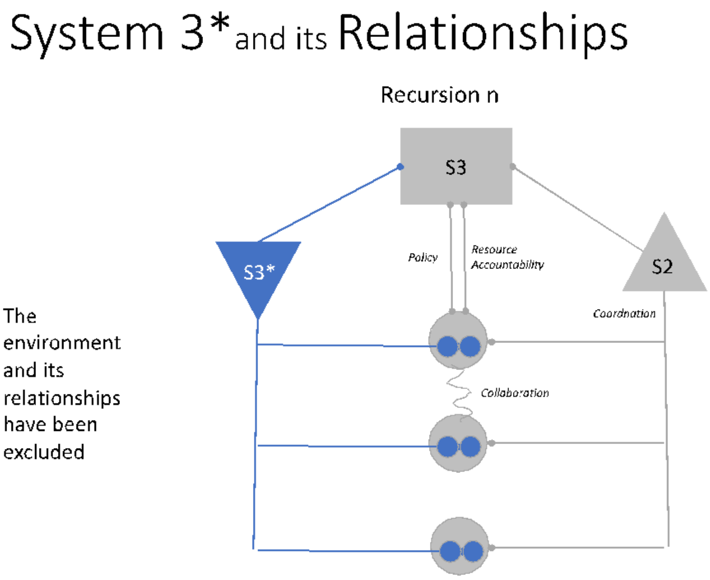

System 3*and its Relationships
Recursion n
S3
S3*
Policy
Resource Accountability
S2
Coordination
Collaboration
The environment and its relationships have been excluded

ILLUSTRATION: SOFTWARE BASED SERVICE COMPANY

In Agile and Lean-Agile as practised at the level of a delivery team there is typically a robust System 3* in the form of retrospectives, Agile coaching, 'stop the line' practices, gemba walks, and (especially in relation to exceptions such as blockers) visual management.

Some of these approaches also work at the Team of Teams level, but are far less regularly used and at the Enterprise level there are few effective 3* approaches in common use.

FINDINGS

The need for System3* practices is usually not formally recognised. Even auditing, designed for that purpose, usually fails to uncover the unexpected.

Where some sort of System 3* does exist, failings include:

- inflexible means such as using checklists or tightly defined standards to try and fulfil this role.
- adopting an intrusive or heavy handed 'audit' mentality that provokes a defensive reaction (leading to concealment of problems).
- preannouncing System 3* visits, or working to a timetable.

QUESTIONS TO ASK:

- Do managers have ways to really understand how their operations work?
- Do teams trust that management understands them?
- Are managers trust grounded or naive?
- How are unwelcome System 3* signals acted upon? Or are they ignored or suppressed?

SYMPTOMS OF FAILURE

High levels of trust reduce the need for disruptive System 3 interventions in the work of System 1. But absence of an effective System 3* failures or shortcomings might not be detected in time, which can lead to:

- 'firefighting'
- a blame culture
- overreaction
- duplication of effort
- micromanagement

SYSTEM 4 – ADAPTING

Collectively Systems 1, 2, 3 and 3* should be designed to manage all activities in the ‘here and now’.

But to remain viable, an organisation must have the ability to change its structures and activities to promote or adapt to observed or anticipated changes in the environment, ‘outside and then’.

This is essentially a process of learning. Some learning is incremental in nature, some requires a ‘step function’ change in what the organisation does or how it does it.

The function of System 4 is to facilitate the codification and sharing of existing knowledge. And it also to anticipate the future and identify what changes might need to be made to meet future challenges or exploit potential opportunities (aka a strategy).

To perform this role effectively, System 4 needs:

- be able to interrogate its perceived or anticipated environment (which will differ depending on the level of recursion) and
- build and maintain a model of its environment and of the viable system in which it is embedded to understand what the implications might be for what is done and how.

Potential is translated into action (or not) through a process of deep and dynamic engagement between System 4 and System 3, the ‘owner’ of design and resource allocation decisions.

Translating perceptions of a changing, unpredictable and often ambiguous external world into practical options for change is complex and difficult. As is the task of making sensible decisions about which of these to pursue and how, given limited time and resources. Also the knowledge required to make good decisions is distributed widely in the organisation. As a result, careful thought needs to be given to how this process is managed and led.

There is often tension between the demands of the present in the ‘here and now’ and the future ‘outside and then’, not least because of competition for resources. And these tensions often cannot be resolved by purely logical arguments. The qualitative judgements needed to make complex trade-offs requires arbitration by a higher source of authority, which is one of the roles played by System 5 in the VSM.

Finally, strategy is part of the role of every viable system at every level, since they face different environmental challenges that are more or less different to those at other levels of recursion. As a result, System 4 work should involve System 4 perspectives from adjacent levels.

Together Systems 3,4 and 5 manage a given level of recursion, as shown below.

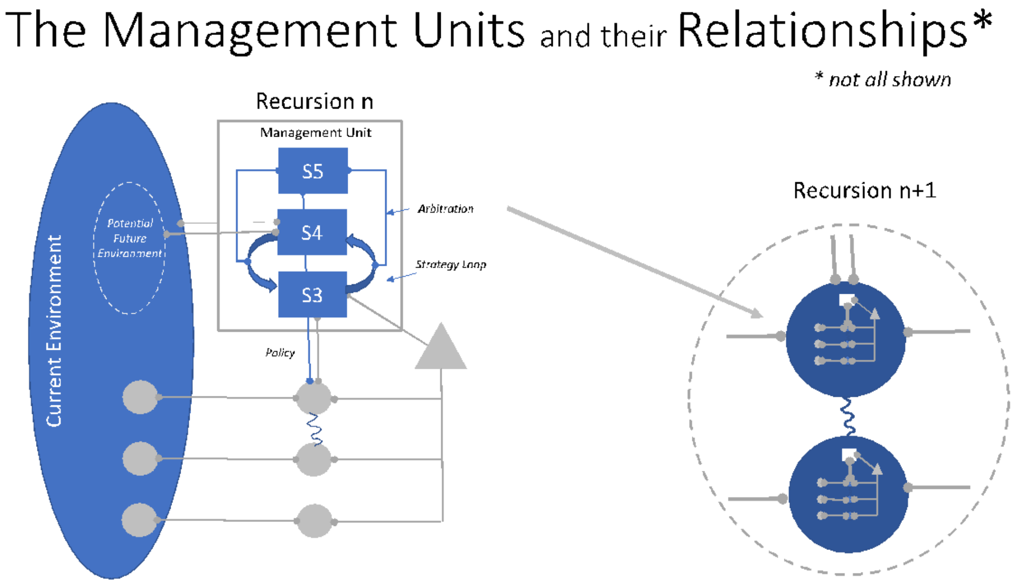


ILLUSTRATION: SOFTWARE BASED SERVICE COMPANY

In an organisation employing Agile methods at the team level, much of the System 4 role is played by product strategy and roadmaps together with feedback from customers or their proxies. It tests and refines the options under consideration, promoting rapid adaptation. Teams have very clear models of how they operate courtesy of standard Agile approaches and take responsibility for their own self-improvement.

At a higher level of recursion, communities of practice (like ‘Guilds’ in the Spotify model) can play a role in promoting incremental improvements. However, in many organisations, ‘strategy’ is seen as the preserve of senior management (i.e. the Enterprise level of recursion) and so is often absent or poorly organised at the Teams of Teams level.

System 4 jobs that should be performed include:

- Researching technological futures
- Developing strategic options
- Skills and organisation development
- Financial modelling
- Market analysis
- Marketing and positioning
- Managing learning.
- Market forecasts

QUESTIONS TO ASK:

- Do we have information we trust about our external operating environment and broader strategic environment?
- Do we have some idea of how those might change in the future?

- Do we have some idea of the scale and rate of change in our operational and strategic environments?
- Do we have a model of what that could demand from us and how that compares to what we are currently capable of?
- Do we understand the strategic risks (existential risks) we face?

SYMPTOMS OF FAILURE

There are many possible points of failure. For example:

- One level of recursion or the whole organisation may have no System 4 at all, thereby failing to remain viable in the longer term
- The strategy of one (usually higher) level of recursions may dominate or overrule that of another leading to disenfranchisement and/or generating activities that are irrelevant or inappropriate
- The balance between the ‘here and now’ and ‘outside and then’ may be struck such that the system fails to adapt or the short term viability is threatened by diversion of too much resource to support speculative or unnecessary or premature activities and ventures
- Failing, or being slow to deliver solutions that add value.
- Delivering solutions which are not consistent with other aspects of strategy (e.g., corporate, technical etc).

SYSTEM 5 – GOVERNING

By definition, an organisation needs to decide what it ‘is’ and consequently the kind of things that it will do and how it will do them – its identity – from a potentially infinite set of possibilities. All and every part of a viable system has an identity, even a team and definitely an individual in a team. And having an identity that is congruent with the rest of the organisation is critical to viability.

Whilst the ‘purpose’ of an organisation or part if it might be written down. But in practice the identity of a viable system (at any level of recursion) is what is seen to be done. In other words it is the product of all the decisions that are made to do one thing instead of another (as in the case of the System 4 and 3 dialectic) or which constrain (or stimulate) actions taken across the entire system.

The latter are sometimes referred to as ‘policy decisions’, which impact the management and operational units.

Collective identity is a powerful cohesive force. So, an organisation with a powerful identity will rely less on formal processes and procedures than one that is more individualistic where the identity of the component viable systems exerts a stronger influence.

The VSM requires that:

- Governance mechanisms be put in place to generate such decisions as and when they are required and that they
- strike a balance between current demands & capabilities and the demands of the future.
- Such decisions are taken in the interests of the viability of the system as a whole i.e. reflecting the interests of all the (sub) systems and the need to maintain a healthy balance between them.
- Governance also needs to ensure the rest of the organisation is ‘in order’ - that it has the structures, relationships and mechanisms to run in a way that complies with the legal, regulatory, societal and ethical norms of the society it is a part of.

Note that the VSM does **not** require that governance be exercised by an individual or a nominated group of senior managers, just that the activity be effectively performed, somehow. For example, democracies are viable systems that govern through an elective system, and an individual's conscience and the decisions that flow from it are further examples of governance at work.

ILLUSTRATION: SOFTWARE BASED SERVICE COMPANY

The small size and extreme focus of Agile teams breed a strong sense of identity and thus internal cohesion, sometimes to the detriment of overall organisational cohesion. Furthermore, the balance between S3 and S4 (overseen by S5) is relatively easy to maintain, since it need only address ways of working, and short-term response to demand fluctuations.

At the Team of Teams and Enterprise levels, S5 tends to be vested in conventional management structures of departments, divisions and the organisation as a whole to take decisions that cannot be taken at lower levels.

FINDINGS

In terms of scaling Agile, decision, policies and the cadence of decision making are often divorced from the operational realities as seen by Agile Teams and the espoused values of the organisation. Our findings show that at Enterprise and Team of Teams levels, there are typically very large imbalances in strategic decision making, so rather than a balance in information and time dedicated to past/current & internal vs future & external concerns, a 70-80% imbalance towards past/internal is common.

Similarly, most organisations fail to spot strategic risks in time and the more successful the organisation, the harder this is.

Relatively few organisations have any assessment of the scale or rate of change in their environment that they need to match to achieve a requisite level of business agility.
The combination of these failures results in the high levels of corporate failure we see: almost 90% of the original S&P 500 are no longer in business.

All these issues can be rectified relatively easily. The research data is strong, showing that the closer an organisation adopts the 'VSM formula' the less likely it is to succumb to an existential crisis[4].

QUESTIONS TO ASK:

- Are actions consistently congruent with stated purposes, values etc.?
- How are deviations dealt with?
- In the strategy process, how are
    - the internal needs of the organisation balanced with the external demands from the environment?
    - the constraints and habits of the past balanced with the possibilities of the future?
- How do we know the organisation is 'in order'?
- How is the right balance between cohesion and autonomy maintained?

SYMPTOMS OF FAILURE

The absence of an effective System 5 can lead to:

- imbalances in decision making - either failing to recognise the need for change or planning a future that is unachievable.
- Internal corruption

[4] Research on 138 organisations showed a .71 correlation between how close an organisation matched VSM and the organisations' ability to survive or avoid an existential crisis. There was a corresponding correlation of company collapses with a failure to match VSM characteristics.

- Lack of organisational ethos
- No mechanism to arbitrate between System 3 and 4, leading to protracted discussion where no resolution is reached.

An ‘over-active’ system 5 can also be a problem:

- As already mentioned, a very strong local identity can result in hostility to those perceived as outsiders by the Team, even if they belong to the same organisation. This might be manifest as resistance to change.
- And it is possible that parts of the organisation who should play a support role, such as shared service centres, develop their own sense of identity thereby pursuing their own interests rather than the organisation they were set up to serve.
- Written statements of purpose and policy, if they are seen not to be observed without sanctions, can have a corrosive impact on organisational cohesion.

COHESION BETWEEN RECURSIONS

As already mentioned, the interface between organisational levels is a point of tension, especially in organisations with Agile teams, since the ethos, mindset and related practices may differ. This is a particular problem as the interface with the enterprise level since this is typically run on an annual cycle, based on top down planning in a way which is antithetical to Agile principles.

There are at least three potential ways of dealing with this potential conflict (which might equally apply to the relationship between the team and portfolio levels):

1. Simply duplicate 'Agile team' level practices at higher levels of recursion. This will not work as the nature and complexity of the work that needs to be done at higher levels is fundamentally different. We call this **'naive scaling'**.
2. Build a firewall that insulates lower level Agile practices from those parts that operate using a different set of rules, particularly at The level of the Enterprise, and find ways to translate from to the other. This means you have levels of the organisation that are relatively rigid and an operational level that is very fluid, so this requires the organisation to run two management philosophies in parallel. Insulating Agile teams from this is difficult and likely to lead to misunderstanding and fractures between levels. We call this '**insulating Agile'**.
3. Because VSM is recursive and uses the same systemic mechanisms and philosophy at all levels, it can be used to integrate Agile at scale. This requires building 'loose couplings' between levels, so that rather than having a rigid structure trying to interface with something much more fluid, there is flexibility across and between levels. We call this '**integrated scaling'**.

Most commentators and scaling frameworks assume that Enterprise level practices are immutable and non-negotiable and so are focussed on *insulating* Agile practices. This paper is focussed on the 'integrated scaling' route and thus the mechanisms that are needed to reconcile the activities of the different levels. These, as mentioned, typically operate on a different frequency with different time horizons, and dealing with different levels of complexity and in response to a different set of environmental relationships.

The nature of decisions taken at different recursions will also differ. In the VSM, it is important that decisions should not be made at one recursion that can be equally as competently made at a lower level of recursion.

Because these differences are not always transparent it is good practice to routinely involve representatives of adjacent recursions in decisions which have a wide impact. Also the interface between different organisational levels is not a single node, as is the case with a traditional hierarchy. Instead, it is a set of relationships between the defined (sub) systems at different levels.

For example:

- SYSTEM 2 for different levels should act to coordinate the coordinating mechanisms at different levels. This may involve timetables, concepts, terminology, cadence, horizons, batches and so on. When and where the need for cohesion is paramount this will involve harmonising on a common set of mechanisms. Where local autonomy is important, as a minimum there needs to be a translation mechanism between the levels of recursion.
- SYSTEM 3 at different levels of recursion need to collaborate to regulate the decision making authority, capacity and the flow and sharing of resources across and between recursion. This too should reflect the desired balance between local autonomy and organisational cohesion.
- SYSTEM 4 should collaborate to share their different perspectives on the outside world to create a set of potential actions that balances directed (top down) and emergent (bottom up) strategies.

FINDINGS

The maturity and acceptance of Agile practices at the level of Teams is not mirrored at the 'Team of Teams' level which, as a result, often adopts the structures and practices of the Enterprise level recursion by default .

Consequently, there is often an unresolved tension between fast-moving externally oriented Agile practices and traditional organisations based on annual budgeting, dominated by an internally focussed culture of negotiation and top down 'command and control' practices. This is because the latter are taken to be a given and non-negotiable, which is not the case. They are simply conventions that have never been challenged.

One of the problems that occurs between levels is that each has a different job to do in maintaining the organisation's 'fit' and relevance to the world outside which is where its value is delivered.

At the Teams level, this means being agile enough to adapt to changing demands on products or services at the operational level, so having the operational agility is key.

At the Enterprise level, it can mean choosing to move away from one market currently being served by Teams and switching to a different market or a completely different offer, so there, strategic agility is key.

The need for operational agility is fairly constant, but the need for strategic agility is much more unpredictable - organisations can run for long periods where their environment is fairly stable and then suddenly the strategic landscape shifts and demands a strategic change. So it's not just that operational agility practised in Agile Teams and strategic agility practised by the Enterprise are different - the relative need for each at any point time can change.

In summary, any approach to Scaling Agile has to be capable of flexing how much agility at what level. This is the connection between Scaling Agile and the wider issue of Business Agility.

QUESTIONS TO ASK

- Are different levels strategically aligned - ie are they trying to achieve compatible strategic ends ?
- Is there a clear link between the direction of change of different levels? For example, is what Teams are doing aligned with the direction the Enterprise is hoping to go in?
- Given that different levels will change at different rates, are the connections between them capable of cushioning the tensions this creates?
- Are there any 'black holes' areas of the organisation where the performance is not understood, or resources have to be scavenged because there is no management structure to ensure Teams get properly resourced?

SYMPTOMS OF FAILURE

A failure to properly align recursions with each other in a way that matches the demands placed on the organisation by its environment include:

- Activities conducted on a cadence that doesn't match that of the environment
- 'Shearing planes' between levels:
    - Teams that don't understand their relationship to the Team or Teams or Enterprise level or vice versa, an Enterprise that doesn't understand the contribution of Teams.
    - One level changing so much faster / slower than another that the connection between them is broken

APPENDIX

Some Typical Co-ordination Issues at the Development Team level – and (System 2) Mitigating Actions

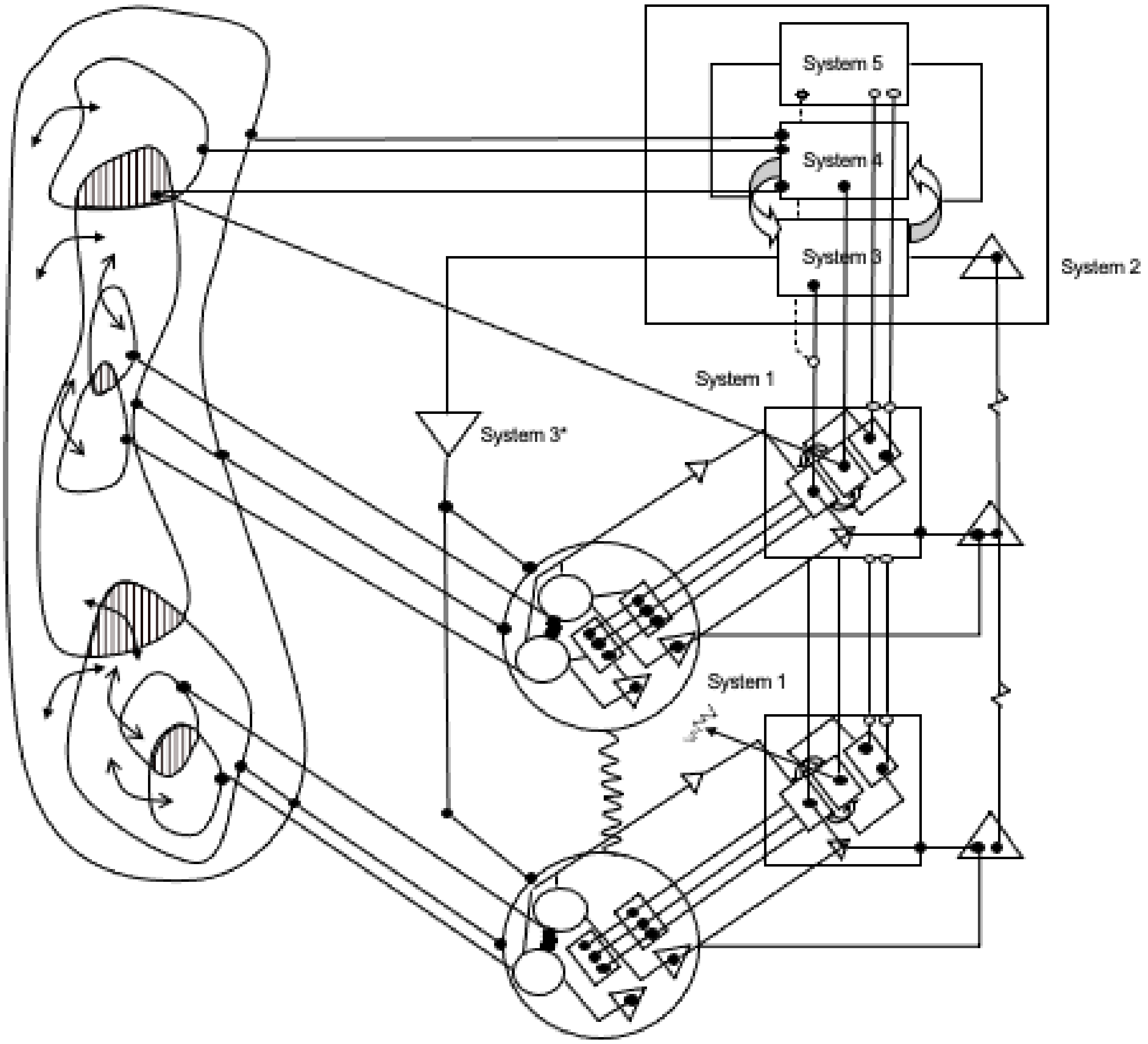
- Sware / functional dependencies
  - Quickly triaging work items for likely dependencies in (or in preparation for) planning events
  - Visually tagging items suspected to have external dependencies, giving them extra visibility
  - Bilateral agreement on protocols for managing affected items (or collections thereof)
  - Sequencing affected items and controlling the number of them actively in progress at any given time
  - Architectural improvements that minimise the impact of dependencies
  - Mock implementations that allow the testing prior to integration
- Convergence on common goal (lack thereof)
  - Ensuring that all planning activities are appropriately contextualised
  - Outcome-based planning and strategy
  - Mutually-agreed measures of success, combined measures of success
- Handovers
  - Look for opportunities to eliminate and replace with collaboration
  - Protocols for conversations at key moments in an individual work item's life cycle (eg the quick "3 Amigos" conversation between Product, Development, and Testing roles just as development starts)
  - Integration, eg between Development and Operations at both process and technical levels, ie DevOps
- Requirements (coherence)
- Tech choices
  - Agreement on authorities for technology choices
  - Minimising the external impact of technology choices – eg integrating via standard or open source protocols that have multiple compatible implementations
  - Choosing technologies appropriate to the domain and expected rate of change (for example trialling new and more rapidly-evolving technologies where a high rate of change is easily accommodated, limiting their use of where stability is valued)
  - Programming language choices matched to past experience and to skill availability, both internal and external

- Operational cadence
    - Synchronising team rhythms (where this helps)
    - Decoupling deployment events, freeing – as allowed by current architectural constraints – different systems to evolve at different speeds/cadences
    - Feature flags / latent code – allowing new features to be deployed in a deactivated form until all dependencies are in place
- Ways of working
- Tools
- System interfaces, APIs
    - Documented (perhaps automatically from the code)
    - Versioned, allowing different systems to upgrade at different times
- Shared code and other development infrastructure
    - Shared code repositories with agreed access rights
    - Agreed protocols (eg code review or pairing with an expert) for changes
    - Continuous integration
- Use of support systems / shared resources
    - Decide whether shared services will remain in place indefinitely or serve as initial enablers to increasingly cross-functional teams (the latter eliminating a dependency but not always economically feasible)
- Skills
    - Encourage intra-team and inter-team collaboration (eg pairing) both for delivery-related reasons (eg to borrow a scarce skill) and for skill development
    - Understand and manage the overall skill profile of each team (more flexible than recruiting only for roles)
- Recruitment
    - Encourage internal recruitment
    - Maximise team involvement in recruitment
- Resource bargaining cadence
- Access to decision makers
    - Manage as one would dependencies; flag any work items likely to require external decisions

Complete VSM Diagram (Three Levels of Recursion)

This diagram is a close copy of the diagram drawn by Stafford Beer, showing all the systems and relationships between them at three organisational levels.

Although it is too complex to be used for practical purposes it demonstrates how the iterative application of a small set of well defined structures can be used to build an organisational model of great sophistication and versatility. It also demonstrates the completeness and integrity of the VSM.


System 5
System 4
System 3
System 2
System 1
System 3*
System 1