

Agile Transformation Journey: Gap Inc. Infrastructure

Executive Sponsors:

Naveen Zutshi, VP Corporate Infrastructure

Champion:

Dave King, Senior Director Infrastructure

Leonardo Murgel, Senior Director PMO

Caleb Wolper, Director PMO

Stephen Bailey, Director Infrastructure Delivery

Mentors:

Jeffery Padgett, Director Infrastructure Architecture, Gap Inc.

Jeff Arcuri, Director Infrastructure, Gap Inc.

Authors:

Kamal Manglani, Consultant, Slalom Consulting

Agile Transformation Coach Gap

Sean Elenz Martin, Solution Architect, Slalom Consulting

Contents

- Introduction..... 3
- Challenges..... 3
- Suitability Filters 3
- Agile Infrastructure Blueprint – a unique framework of practices, behaviors, structure and tools..... 4
- Plan to develop a Blueprint 5
- 10/20/70 Continuous Learning Model 6
- Metrics for the Team 6
- Promoting T-shaped skillset development 7
 - Visualizing the Work in Progress 8
- How to Scale Agile across the Infrastructure Portfolio? 11
 - The new driver: Survivability 11
- Scaling Principles 12
- Tools and Maturing Automation Capability..... 12
- Configuration Management, Continuous Integration and Release Management 14
- McKinsey 7 S Model..... 15
- Value Stream map to improve Infrastructure Service time..... 16

Introduction

Traditional Infrastructure is dead. The traditional Infrastructure Sys Admin Role is dead. The traditional network engineer is dead. Traditional capacity planning is dead. All Infrastructure engineering is software development now. The paper describes Gap Inc.'s IT Infrastructure Agile Transformation journey using an Infrastructure transformation blueprint that suggests an agile mindset model, an agile framework based on lean principles, metrics, behaviors, tools and team structure. The Agile for Infrastructure blueprint is a homegrown transformation roadmap developed at the Gap having its foundations on Agile Principles.

Challenges

Infrastructure organizations face the following challenges when it comes to delivering Infrastructure projects using operations resource.

- Lack of visibility into Infrastructure assets & their performance
- Un-predictable deadlines for Infrastructure projects
- High amount of waste due to context switching
- Extended wait times
- Interrupt driven work culture
- High dependency on vendors due to outsourcing of 80% of technical workforce.
- Rigid strategic outsourced vendor contracts focused on very high service levels rather than driving technology best practices

Suitability Filters

It is difficult to apply agile principles directly in the Infrastructure space since many of the suitability filters such as Incremental delivery, single goal, team stability or dedicated resources, collocation and failing fast are not directly applicable. This becomes even more complex when the structure of the Organization is matrix based with extended decision making timeframes. The challenges faced by Infrastructure projects are very different from traditional software development and this area has proven to be problematic for most organizations who struggle with the natural points of friction in trying to adopt agile practices for infrastructure.

Filter	Importance	Pre-Transformation	Post Transformation
Colocation	Medium to High	Very few teams co-located	TV Screen & Live screen share applications will improve co-ordination.
Nimble Teams	High	Very few teams functioned towards set priorities	Formation of PODs as mini Infra organizations self-sufficient to work on almost all DevOps work
Organizational	High	Keen desire to	Breaking down silos

Maturity/ Decision Making		streamline decision making but limited due to heavily matrix structure	using the POD to improve learning and sharing across teams.
Tools	High	Roadmap to upgrade / bring in new tooling	Named Technical communities on specific products
Personal Growth	High	Out-dated training programs, processes, technologies	Improvise learning sharing through on the job coaching. Implement 10/20/70 model
Dedicated Resources	Medium	Out-dated team structure leading to heavy multi-tasking and context switching	Draw a focus to work in progress. Create a visual depiction of all work in progress using JIRA. Plan capacity based on WIP limits and organizational drivers
Work In Progress Limit	High	No established limit visible. Target resource utilization >100%	Identify capacity needs based on needed velocity.
Agile Mindset	High	Desire to be agile	Keep the mindset of continuous improvement
Outsourcing / co-sourcing	Medium to High	Highly outsourced with less core skills retained	Build core skills internal to organization
Executive Sponsorship	High	High executive sponsorship in being and doing agile	High executive sponsorship in being and doing agile
DevOps evangelist	High	Infrastructure visionaries & thinkers	Continue building Infrastructure visionaries & thinkers

Agile Infrastructure Blueprint – a unique framework of practices, behaviors, structure and tools

The goal of the blueprint is to align formalize practices in the Infrastructure space with the agile manifesto and solve the above challenges. The blueprint proposes using agile practices in the infrastructure projects. For e.g. writing a story in the infrastructure world follows the same format as in

software development however here the stories are written by the engineers in lieu of the product managers and technical managers. The value demonstrated in the story maps to the end vision of the infra project.

The blueprint dives deeper into the behavior and structure of agile team and how we encourage T-shaped resources by forming PODs (basic Infrastructure unit to perform a minimum set of Infrastructure responsibilities). The transparency is brought in by using JIRA as a tool for Agile Project and Story Management along with a unique way to run crisp stand ups (using JIRA Rapid views – an Industry best practice) respecting most infrastructure team members have very isolated responsibilities however providing them improved opportunity to integrate together as a TEAM.

In addition to Agile practices the new process uses Kanban lean principles to bring in transparency and also bring in higher efficiency. At its core the team will pull work into its work in progress queue rather than await assignment of work, thus reducing the management overheads and also encouraging self-organizing capability.

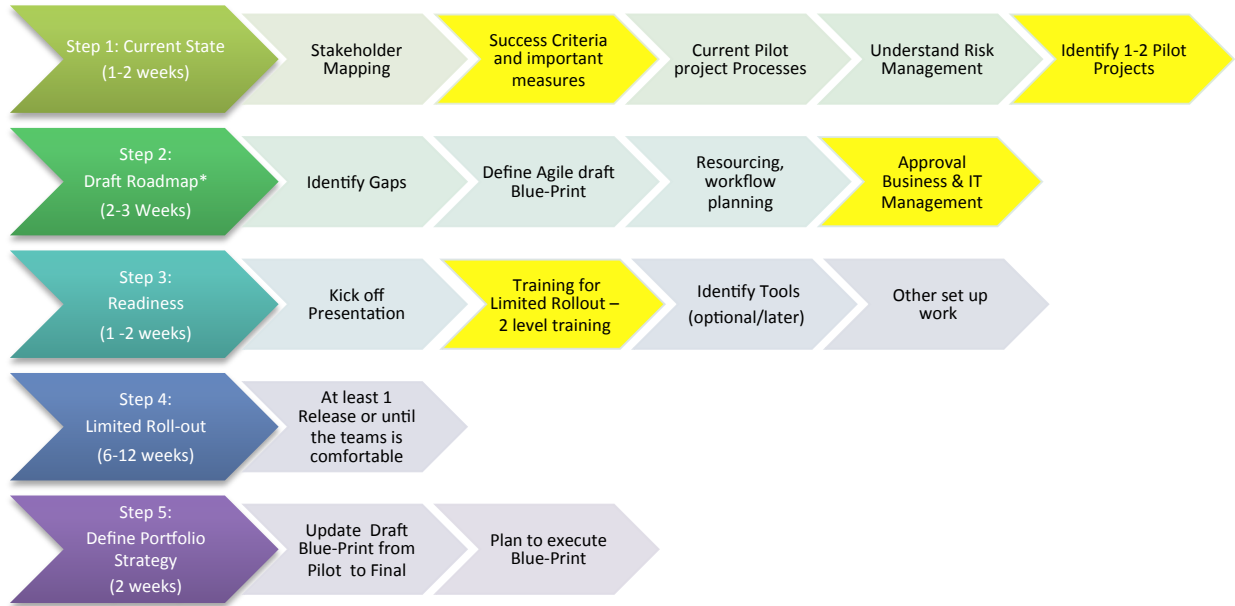
The focus of the blueprint is to reduce the amount of work we are doing and to increase the amount of work that is being completed by reducing context switching and improving focus.

Plan to develop a Blueprint

The development of the blueprint and its success is based on understanding the organizational culture. Once we have the unique ability to meet the organization where it stands in its journey we are in a better position to adopt agile. The process starts with a due diligence exercise which attempts to identify pain points that need the transparency so that the blueprint supports a process to solving them. Second step is to identify champions or pockets that can understand the key practices and become sparks to people who are driven by their influence.

Performing 2-3 pilots to ensure the blueprint is highly tuned to the organizations needs and challenges usually tests the success of the approach. The plan below denotes the approach on getting a blueprint development process initiated. The blueprint will be a living document connecting strategy with execution agility.

The major un-lock was shared ownership and had contribution across stakeholders.



10/20/70 Continuous Learning Model

The training was organized in a continuous learning model to speed up adoption.

10% of the learning was through classroom training through a 4-hour session with the project teams.

20% of the learning was from observing peer-to-peer coaching, feedback from other agile teams, brown bags within communities focused on role transformation.

Lead by example and observing more mature teams.

70% of the learning was by doing & on the job experiences.

A highly effective Agile Transformation coach supported all 3 levels of adoption.

Metrics for the Team

Agile Metrics are not just for management reporting but are mainly for the project teams to drive behavior change based on the new agile framework.

Metrics that help individual contributors within teams understand their commitment to teams' performance and hence to the organizational goals.

Metrics	Description	Frequency
Commitment Ratio	#Stories moved to in progress during iteration/#Stories in Backlog at Iteration open	Iteration
Execution Ratio	#Stories moved done within an iteration/#Stories in progress during the iteration	Iteration
Program Burn up chart	Graphic view of expected time to complete program backlog	Iteration
Iteration Burn up chart	Graphic view of committed story points versus daily progress.	Daily
Average Velocity per FTE	Team Velocity/ Total FTE count	Iteration
Cycle Time	Longest period of time taken to complete a story within an iteration	Iteration

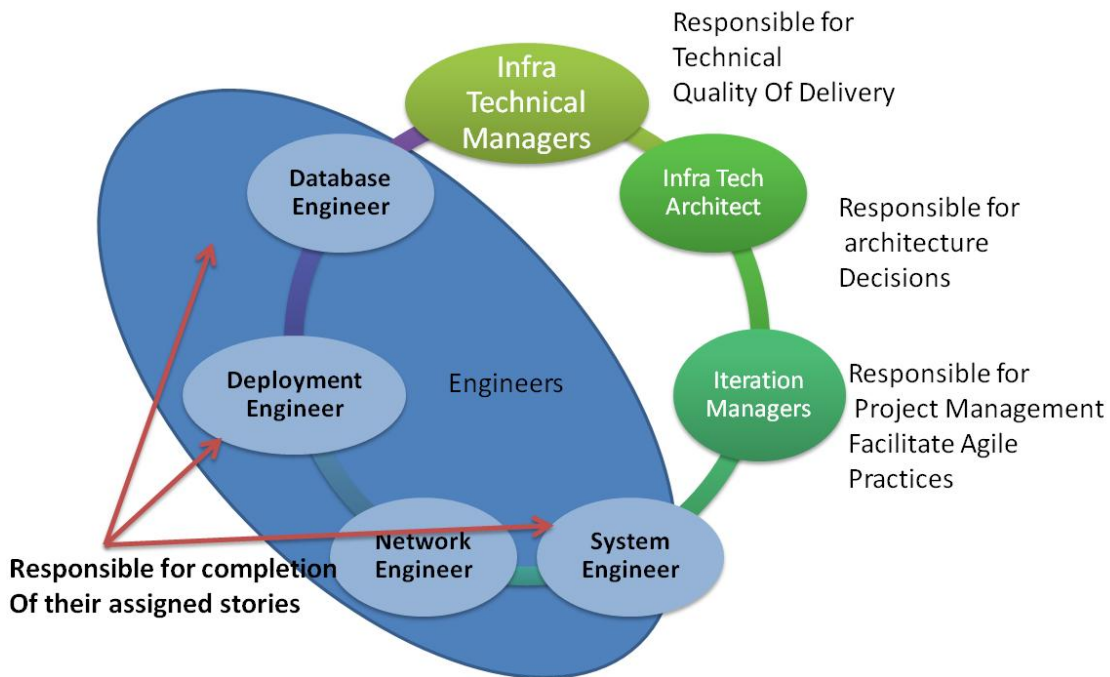
Promoting T-shaped skillset development

The essence of agility is the cross functional aspect from the teams. The goal is develop a team that has lower dependency on external stakeholders. This helps the team to reduce its time to market and have a leaner approach while delivering high quality products.

Building of cross-functional teams must be responsibility of every team manager who must move beyond ring fencing or vertical-ization of resources (e.g. managing a team of Oracle DBAs only).

A better approach would be to identify the minimal unit of team required to perform 80% to 90% of teams' task within the team. This is also called as building of a T-shaped skill set.

Today's large sized corporates operate in a shared service model where specialized resources are pooled together and operate only on ticket-to-ticket basis. They often lack the overall vision or objective of initiatives an organization is in progress of developing. The leadership and guidance comes from 3 primary roles: Infrastructure Technical Manager, Technical Architect, and Iteration Manager. The Infrastructure Technical Manager role is mapped to the product manager role in the software development world. The Infrastructure Technical Architects provide the architectural vision and deep technical solutions. Finally, the iteration managers, who act similarly to the scrum master in the software development context, round out the core team.

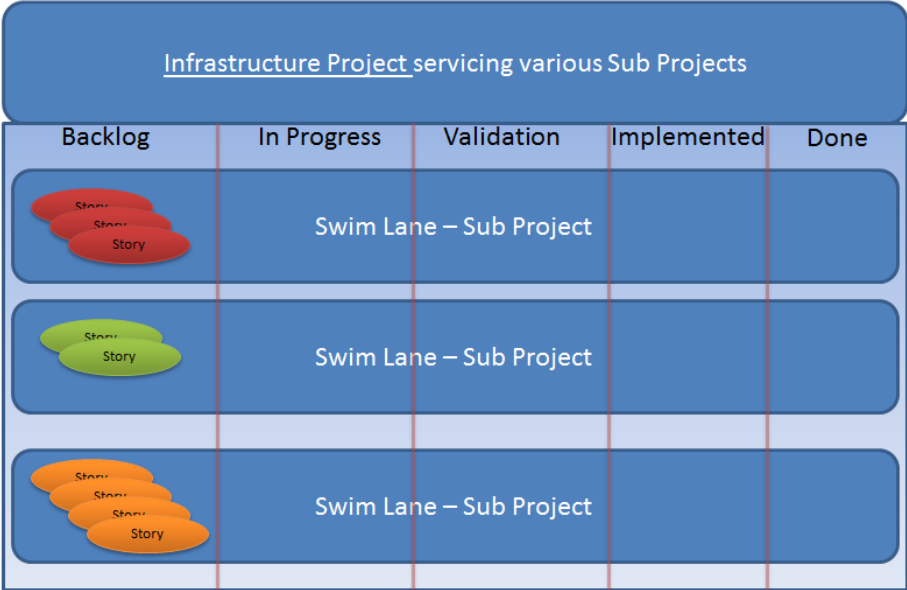


Building a cross functional unit as above helps identify common cross-training of tasks that are needed by almost all projects (creating URL, configuring IPs, ports etc.) which are also a cause of bottleneck if the team was dependent on getting them done by specific resource. This helps build a cross functional skillset ***which improves productivity by over 30%.*** Apart from this the team also starts working together as an infrastructure mini organization for an application development project which is perhaps a very powerful start up feature of a cross functional POD as above.

Visualizing the Work in Progress

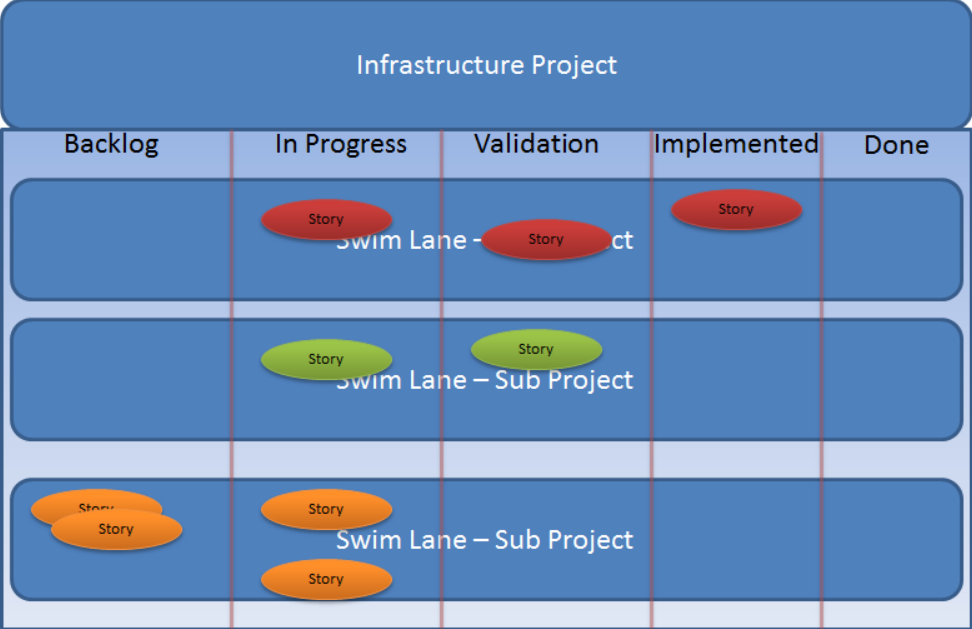
One of the most adopted agile practices is the standup meeting. The challenge with Infrastructure teams is that they work on multiple projects at the same time and individual resources work on projects that are not directly related to their peers. *For stand ups to be successful they must be crisp, quick and accurate.* The proposed format is a visual of all the work being done by the team on multiple projects by a single infrastructure team. This visual provides a view of how stories navigate from backlog to in progress to validation to implemented to done. A starting visual view at the beginning of the iteration is shown below.

Rapid View Stand ups with Team



Once the iteration is kicked off the team starts working on stories and delivering the work. The board keeps showing the real-time status of the work similar to a production line up for manufacturing.

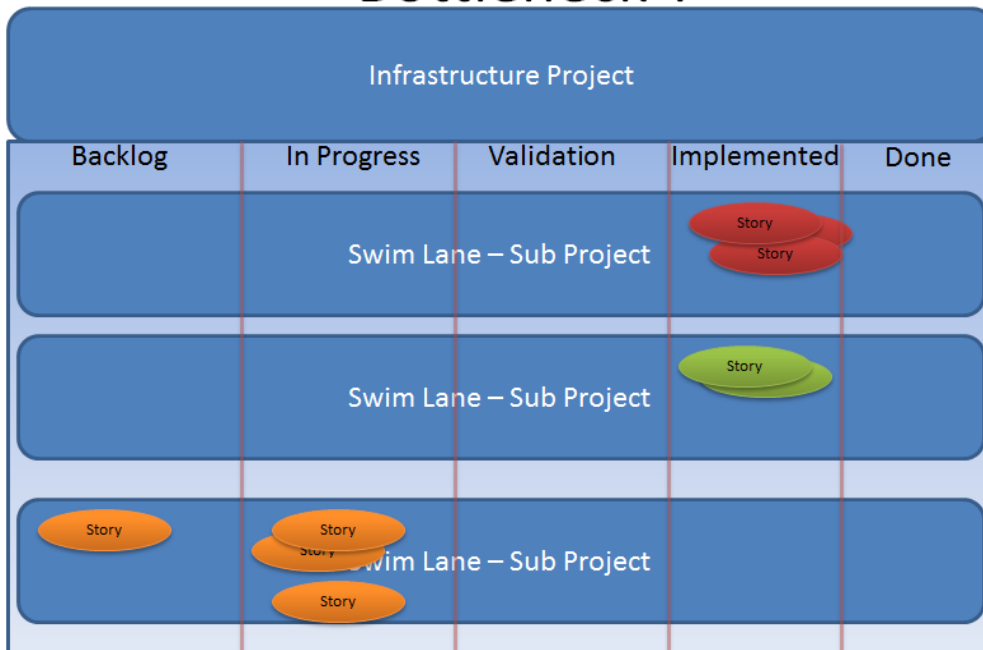
Core Concept – Rapid View Implementation



Now say we observe queuing up of stories in particular project within the in progress queue. That is a smart indication of a bottleneck that needs immediate response by the team to ensure that commitments are met.

At times to improve productivity the format of the board may need to change. This is to specifically ensure the bottlenecks have more visibility.

Core Concept – Rapid View Implementation – Seems to be a Bottleneck ?



It is ok to modify your rapid view or Kanban board to analyze/solve a bottleneck

Tools that provide online real-time white boarding visual of the work are very limited in the market. One such tool is the JIRA Greenhopper product. Typical timelines for adoption of a tool such as JIRA are between 3 months to 9 months. The primary challenge is the resistance to increase transparency.

How to Scale Agile across the Infrastructure Portfolio?

Agile is mainly about building leaders in terms of empowering teams to make decisions. The underlying philosophy is having confidence in your teams which sends reassuring signals to the team on a continued basis and helps them to correct the decisions quickly when they have gone down the wrong path. The leaders build a value of failing fast which is to try and work through team decisions but at the same time have corrective measures in place to quickly correct course of action. This results in early launch of new products without wasting millions of dollars on expensive decisions that are implemented and later result into failed product launches.

The new driver: Survivability

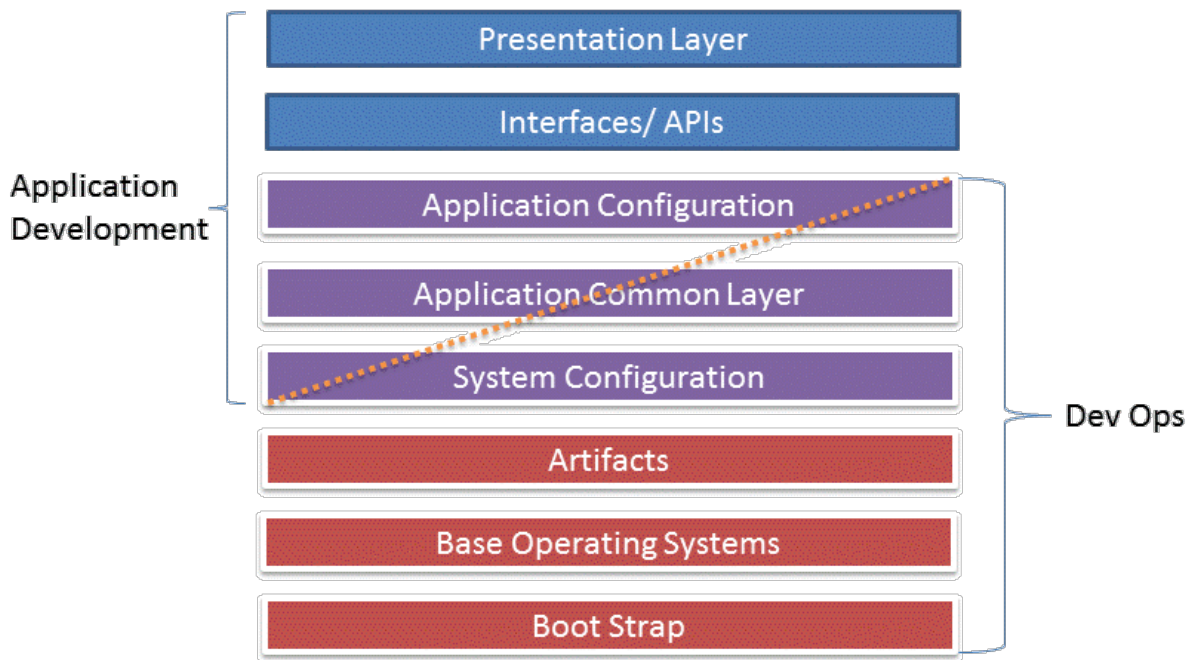
The agile adoption is no longer a choice but a question of survivability of various organizations. Professor Robert Sutton at the Agile 2012 communicated 7 Scaling Principles that leaders need to focus on when aiming to change the culture using agile.

Scaling Principles

- **Link hot causes to cool solutions:** continuously communicate the cool achievements or differences of teams that demonstrate breakthrough performance. *“Mingling’ of teams is an automatic process for spreading agile mindset.*
- **Live a mindset, don’t just talk about it:** Invest in training efforts on the mindset and building of cross-functional skillset. *Support a culture of fail fast and rapid corrective action.*
- **When in doubt take out cut cognitive load:** keep the blueprint simple and tune the language for the audience. Vague words cause more harm than helping the transformation. Key observation is to have a very lightweight tool, lightweight process on the tool and a simple vision that people use to solve difficult problems on day-to-day projects.
- **Use little things that pack a wallop, the power of subtle cues:** Traditional Infrastructure teams are driven by interrupt driven work and a continuous change in priority. *Implement a limit on the work in progress and while an engineer is busy playing a story ensure he or she is not disturbed as this causes a lot of waste in terms of context switching and reducing quality.* A good agile leader shields the team of interruptions and keeps their WIP limit under control.
- **Connect people and cascade excellence:** Pilots are the best way to test the blueprint and refining it. Cascade the practices gradually until the entire organization has learned to adopt it. Good leaders and coaches know it takes a while and sometimes a few years to bring in culture change into a large corporate organization.
- **The mindset is the steering wheel incentives are the juice:** *Continuous encouragement is more important than only setting a goal in the incentive systems.* Focus on practices such as retrospectives, which are purely driven by metrics to ensure continuous improvement.
- **Don’t put up with destructive beliefs and behaviors:** When we do not make tough decisions we do not use agility to its full benefits. *Every time an agile transformation has failed it demonstrates a very rigid culture of a company with a very low attention to market priorities.* If there are issues in agile adoption after the first few iterations deal with them along with the team decisions. This instills confidence in leadership and openness on behalf of employees to come forward and provide valuable feedback.

Tools and Maturing Automation Capability

Infrastructure now is represented, configured, and deployed as software code. The advent of on demand infrastructure, also called "cloud computing" is one reason for this transformation, but not the whole picture. Software development practices have evolved to meet increasing demand for quick deployments. The ability to "fail fast" is turning alpha and beta software code into production code for new products. Making production changes live requires that software developers use continuous deployment methodologies. Continuous Deployment and Continuous Integration software development is tightly linked to agile software development practices. *As corporations begin to adopt fail fast development to improve time to market, the need for adoption of CI/CD with Agile will follow. This transition, coupled with the use of more on demand/ cloud infrastructure offerings further envelops infrastructure as code requirements. Thus, begins the convergence of infrastructure with software development.*



The diagram above indicates the relationship between the traditional application developer and the Operational Developer (DevOps). The layers of interaction described in purple show the required collaboration to ensure a successful process automation project. Developers may require specific components at the system configuration (libraries, jvm...), application common (containers such as tomcat and jboss) and application configuration (xml and properties files) layers, however DevOps is responsible for ensuring the consistent, reproducible and stable deployment of all of those resources via complete end to end automation.

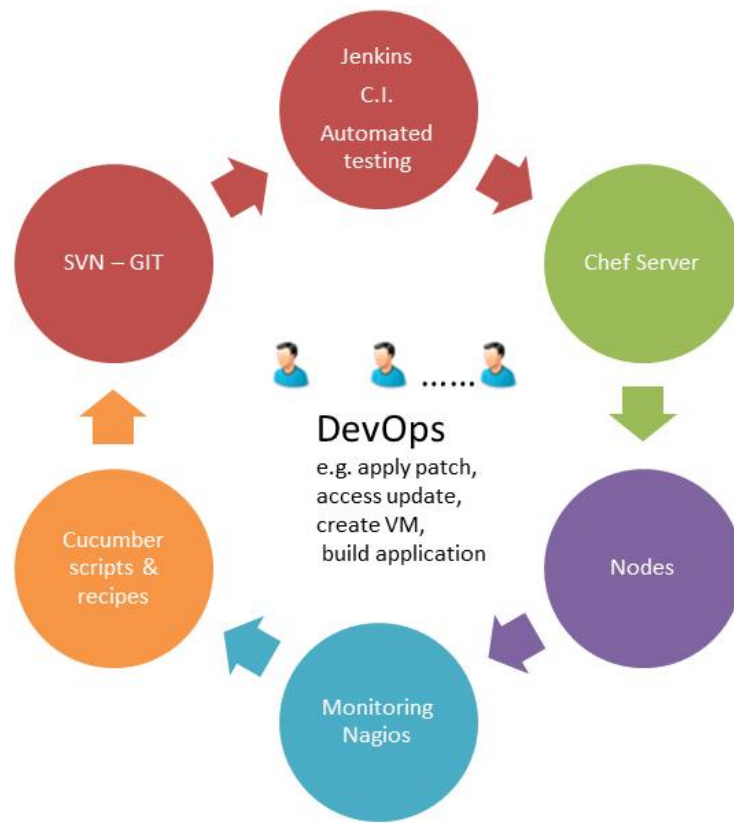
Additionally, automation of provisioning and deployment introduces the ability to rapidly respond to changing business needs across a sea of geographically dispersed physical and virtual resources in an effective and efficient manner. In traditional shops, changes to development and production environments, including acquiring systems, took weeks to months. Modern changes can be implemented in minutes to hours. This reduction in costs is also augmented by the ability to rapidly adopt new technologies that produce results at a better price. Ultimately benefits to the business can be measured in terms of money, time, and resource management.

Configuration Management, Continuous Integration and Release Management

Managing, designing, deploying and testing infrastructure is now in the same fashion as iterative development in software. In the traditional infrastructure world a change to access policies would need a few weeks to be completed based on the number of servers the change would need to be executed on manually. Today, with virtualization, it is easier to manage virtual machines compared to standalone servers as similar policy updates could be done in a shorter time frame of less than a few minutes. These changes are carried out through standard “cookbooks” using Opscode Chef for configuration management or similar tools like Puppet.

The code that builds deploys and tests our infrastructure should be committed to source-control such as GIT. Tools like GIT and Chef allows the treatment of what are traditionally considered “infrastructure” parts of the system as code. With Chef one can script and deploy VMs and Operating Systems along with their configurations as well as the applications that run on them. It offers a near complete provisioning ecosystem through code.

Continuous Integration (CI) utilizes tools like Jenkins to automate testing and advancement of application code and also infrastructure code. This also leads to improved quality, consistency and supports Infrastructure defect management to prevent issues such VM crashes, unwarranted differences between environments.

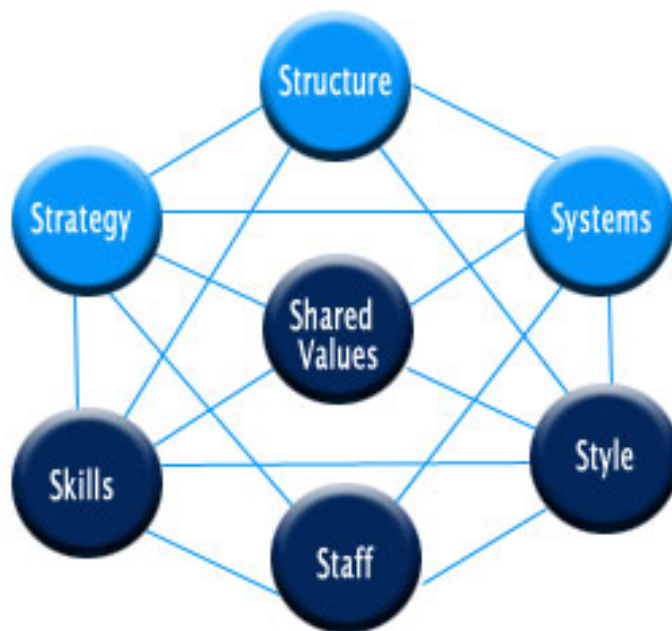


Once the C.I. is complete the next step that we plan on automating is continuous deployment. Monitoring of our services in production is done using Nagios and monitoring scripts are written as Chef Recipes.

McKinsey 7 S Model

The McKinsey 7 S Model is a good model to showcase what elements need to change to make a strategy successful in other words what parts of strategy need to be managed to make change management successful

Structure – Adopt a cross-functional structure to increase independence of teams and form a leaner organization focused on doing less but getting a lot done.



Systems – Adopt a tool to make visible all work being done, show all bottlenecks, and drive the decision-making process using metrics. Subsequently make changes to strategy and observe change in metrics. Design systems to show work in progress limits.

Skills – Develop & support T-shaped skill set.

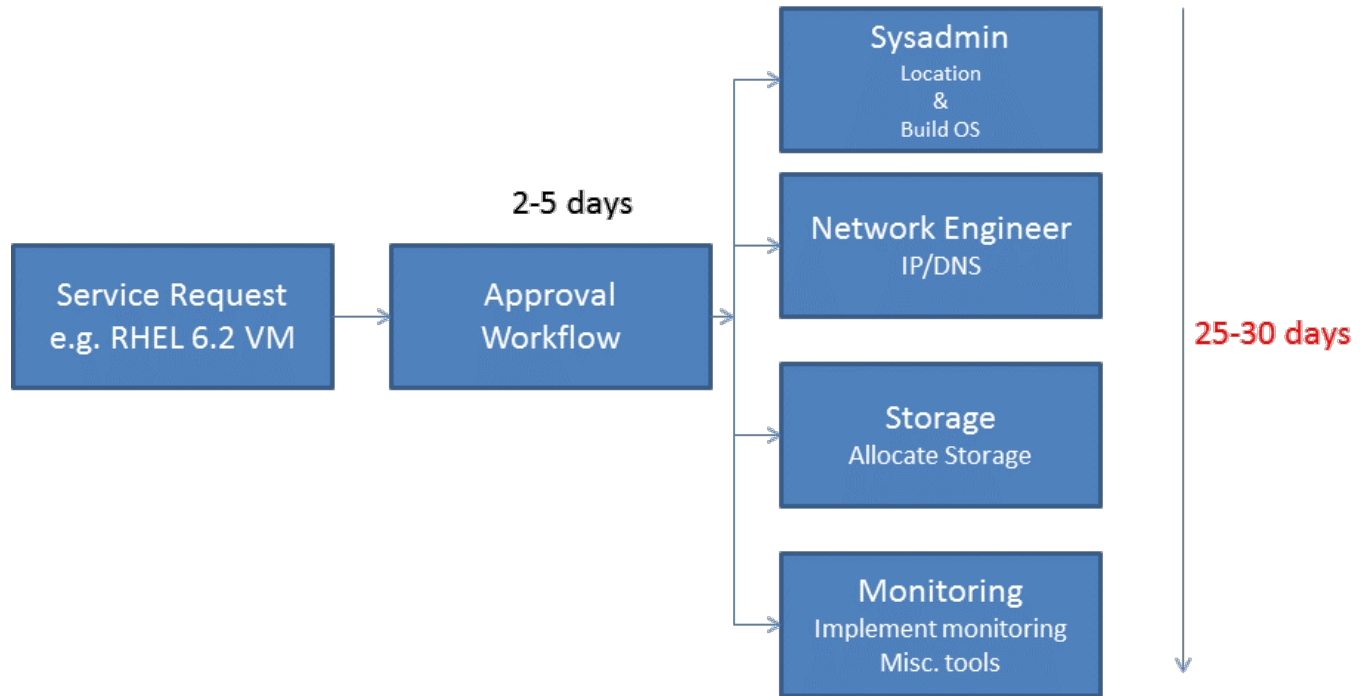
Staff – Listen to feedback from staff & teams and take prompt actions to show responsiveness during retrospectives. Invest and Build teams to have a well-balanced skill set.

Style – Leadership style demonstrates confidence while at the same time not really sure of the outcome. Demonstrate openness to change and fast action on bottlenecks or issues.

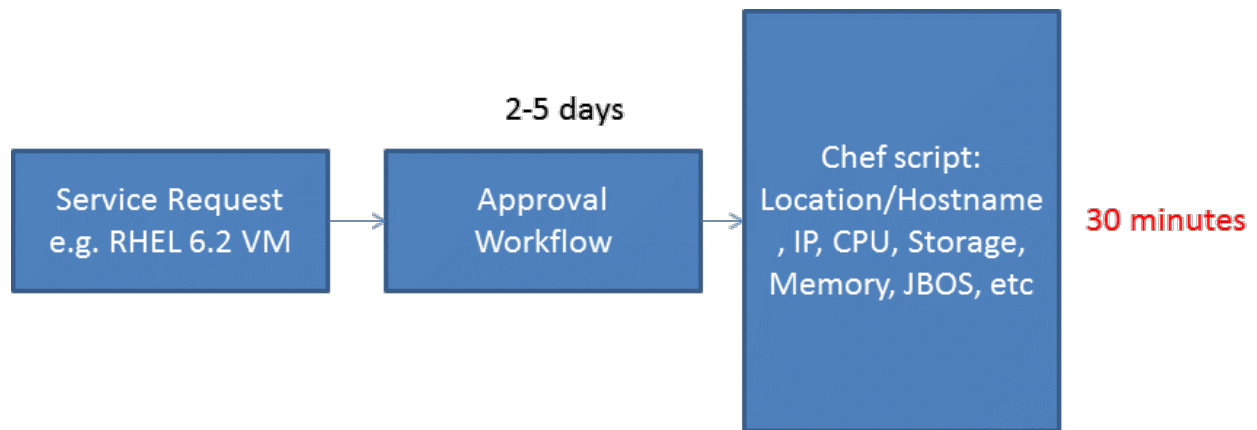
Shared Values – Manage change as a cultural change more than a change in IT processes. Have slack to innovate in business as usual processes. Do not utilize resource at 100% or more of the capacity.

Value Stream map to improve Infrastructure Service time

Infrastructure organizations are plagued by a huge turn around times that throw major initiatives off track. This stalls the growth of an organization. Imagine a development team needing a Red Hat VM created for development work. This work traditionally took up to 35 days and at times even more if there are defects. Drawing a simple value stream, time to value is 35 days which is not good for today's rapidly changing enterprise.



The same service request can now be cut short to a more manageable time frame due to advent of automation and configuration management tooling taking our datacenter management to the next level.



The long pole of waiting for a sysadmin, network engineer, storage engineer and monitoring to their parts is now fully automated through Chef. This cuts the manual deployment times from 30 days to about 30 minutes. The approval cycle is the same. The time to value for servicing infrastructure service requests has dropped by 80% of initial required time along with tremendous savings in human resource costs.

Acknowledgements:

This paper acknowledges efforts from the following leaders at Gap Inc. who have supported the Infrastructure Transformation at Gap Inc. They include: Jeffery Padgett, Director Infrastructure Architecture; Caleb Wolper, Director PMO; and Julie Salazar, Project Manager.