

FOCUS ON END TO END FLOW

Visualize work & flow using Kanban boards and Flow Diagrams

Manage and Improve Flow

Limit work in Process to accelerate Cycle Times and increase Agility



Stories level board & flow

Try Feature/Service/Product
oriented
BizDevTestOps Teams



Why wait? Bring feedback to the front!
Test/Ops Aware Development

Use Smaller integrative valuable slices of work

AMPLIFY FEEDBACK

Bring Dev • Test • Ops to work together all the way

Bring Ops feedback up front (Operations Driven Development)

LOOPS

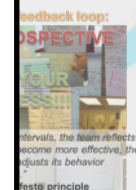
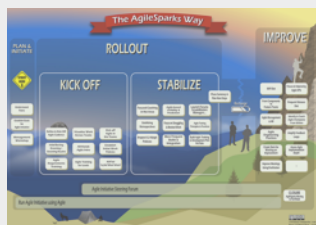
Increase frequency of Delivery



DevOps – the future of Agile – why, what, how?

Yuval Yeret
AgileSparks CTO
yuval@agilesparks.com
@yuvalyeret on twitter

Slides available at:
<http://bit.ly/AgileLL4-DevOps>

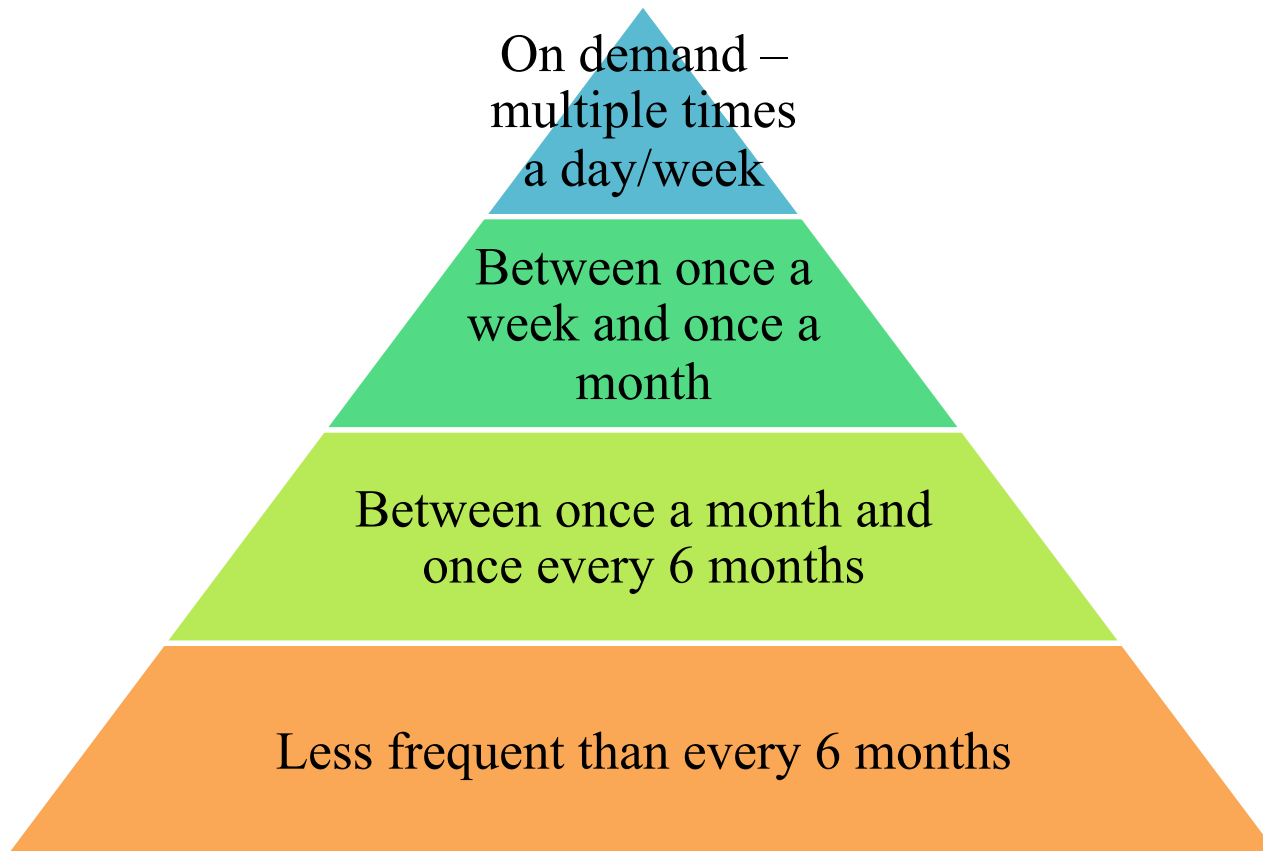


ive/focus

what hurts

DEVOPS

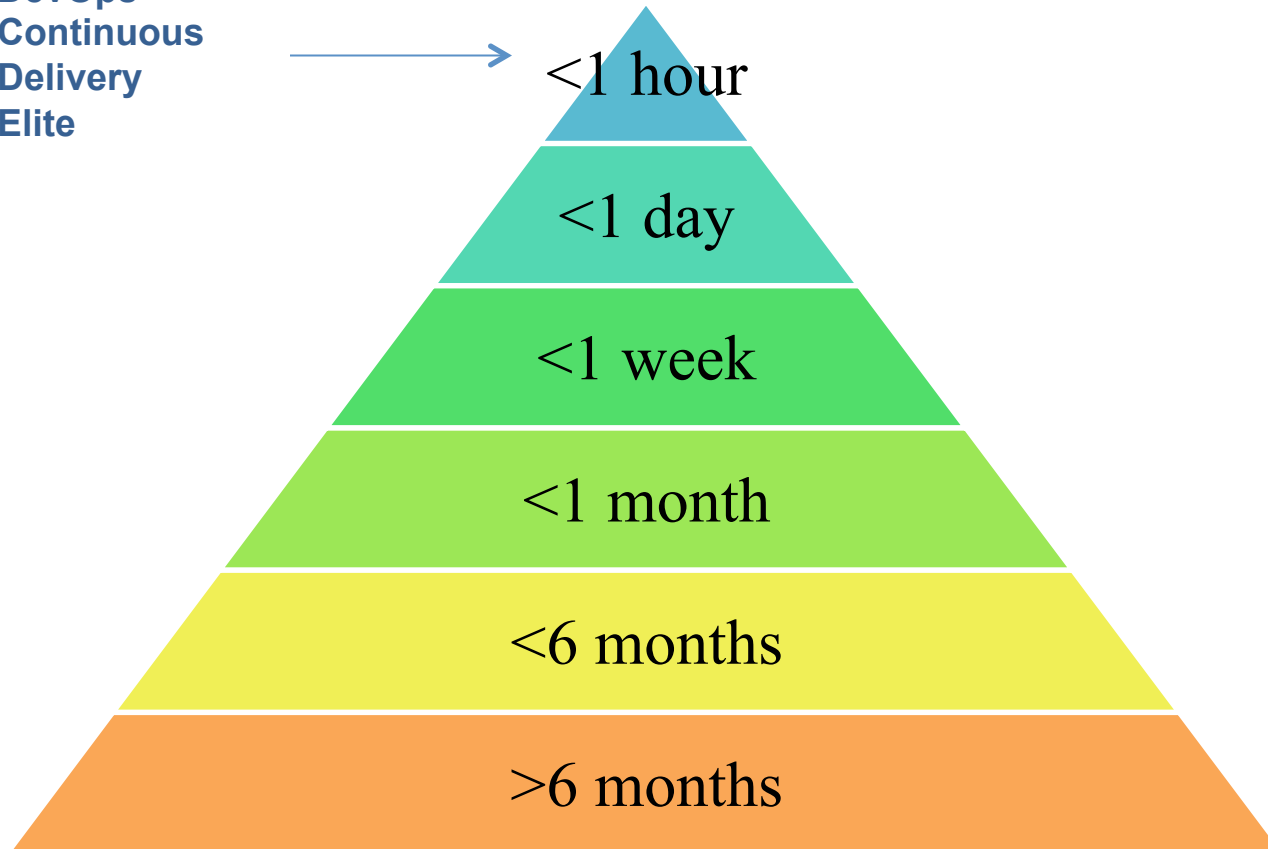
- How frequently are you deploying to production?



Extend Breadth

What is your Change Lead Time - How long does it take for a small change to go all the way from start to production?

DevOps
Continuous
Delivery
Elite



The business winds shift faster and stronger each year



http://www.faculty.umb.edu/gary_zabel



When do you celebrate/relax? Handoff or actual finish line?



Complicating things is often faster in the short run



So you spend most of the time **FIGHTING** fires

"The RIGHT thing" is a hard to nail moving target



"I'm losing sleep over whether we are actually doing the right thing in this big project and not wasting our time" IT Director in a major telco

Familiar?



So don't be surprised if people optimize accordingly



Harder and Harder to deliver

Deliver lots of value

Minimize disappointment/surprises

The Expectations

Be Flexible Enough



"I call this the **IT downward spiral**"


Gene Kim



The IT Applications Development and Maintenance Expectations & Landscape



Deliver more
with less

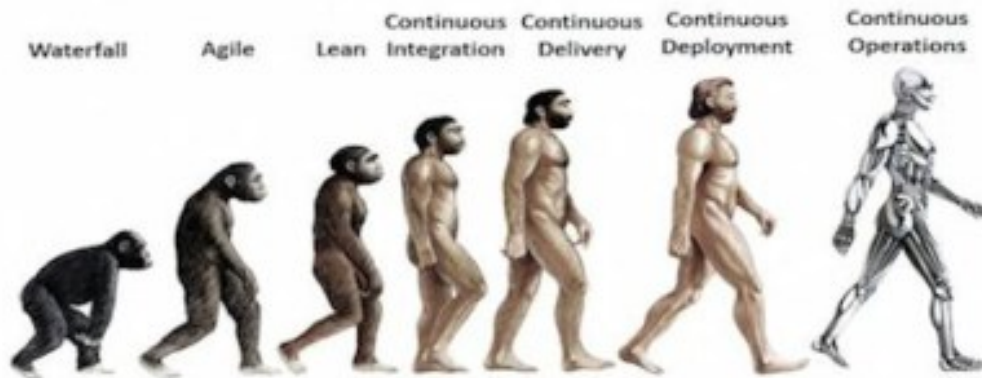


Achieve high
uptimes with
less

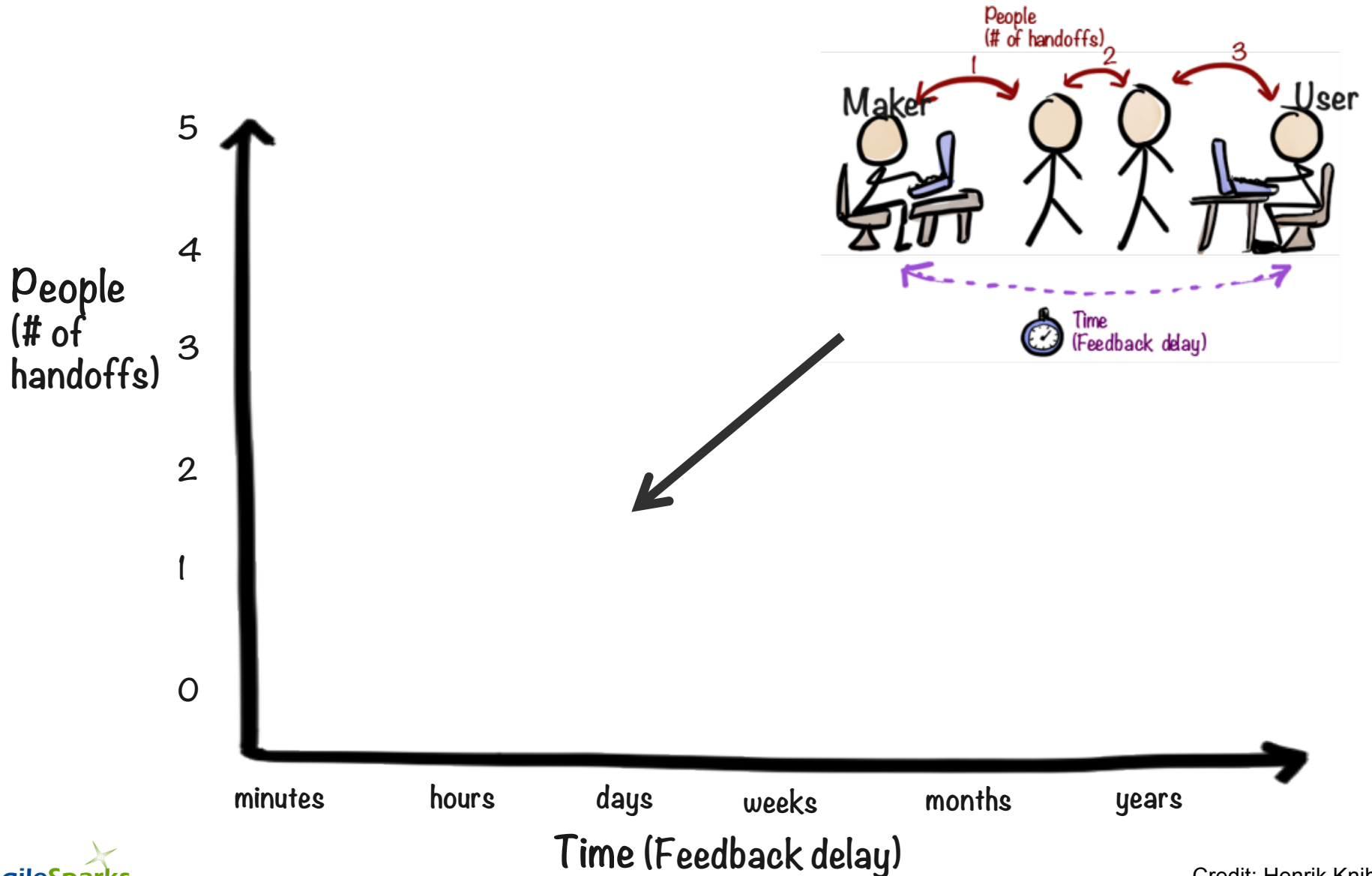
DEVOPS

Aiming to break the legacy conflict between Dev & Ops by leveraging lean/agile thinking/practices at a wider scope than just the development

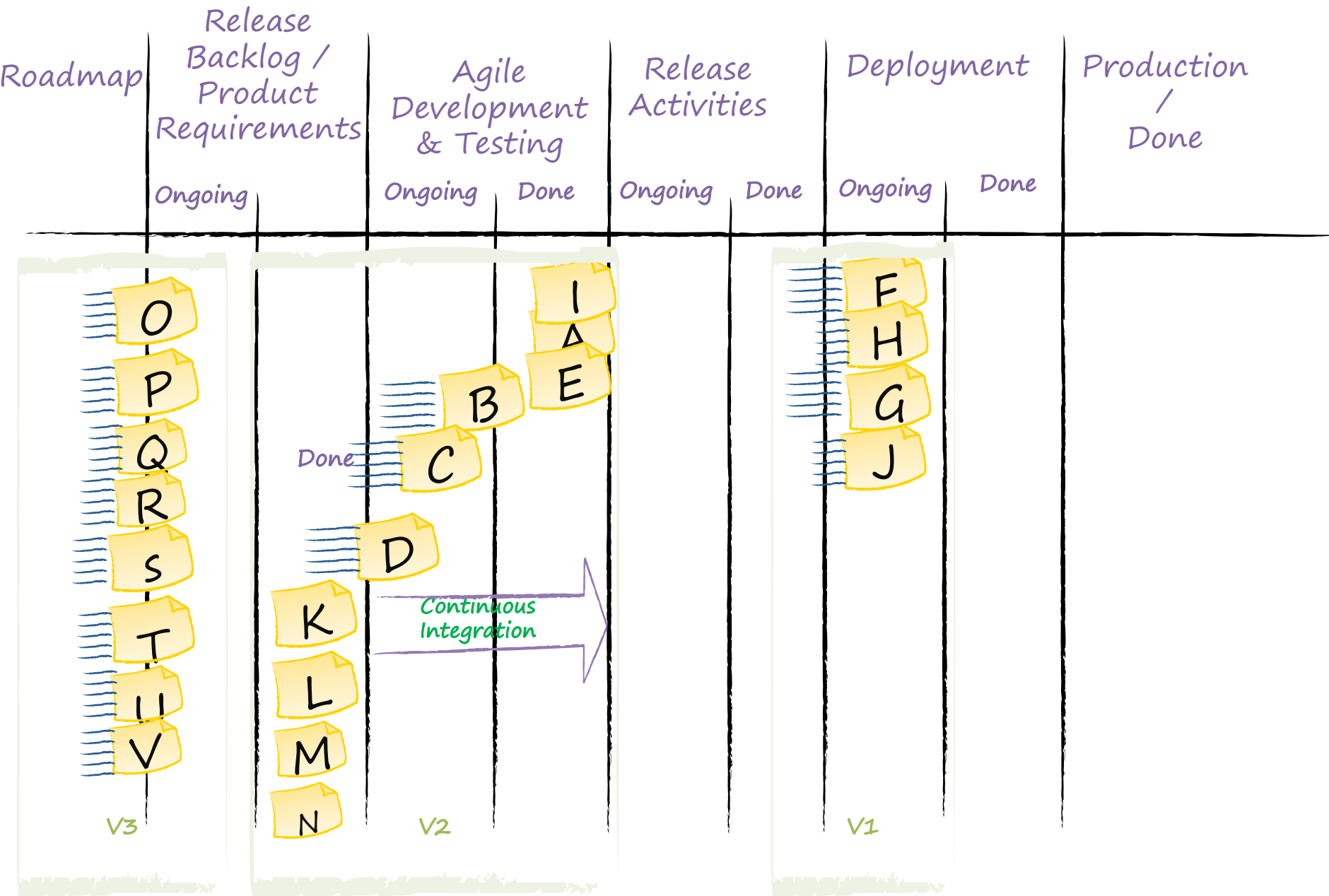
DevOps Movement



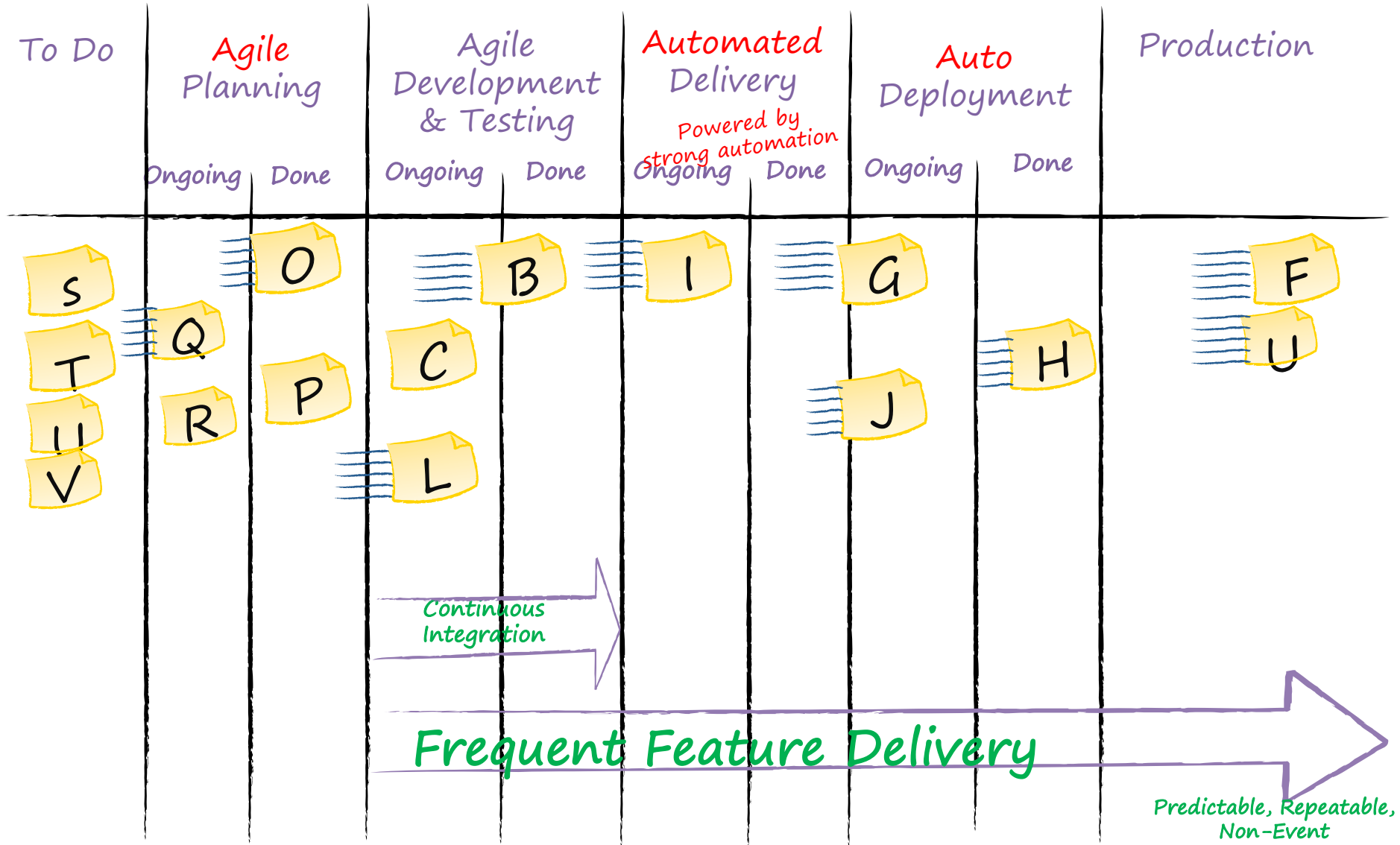
DevOps – Minimize Distance from Maker to User



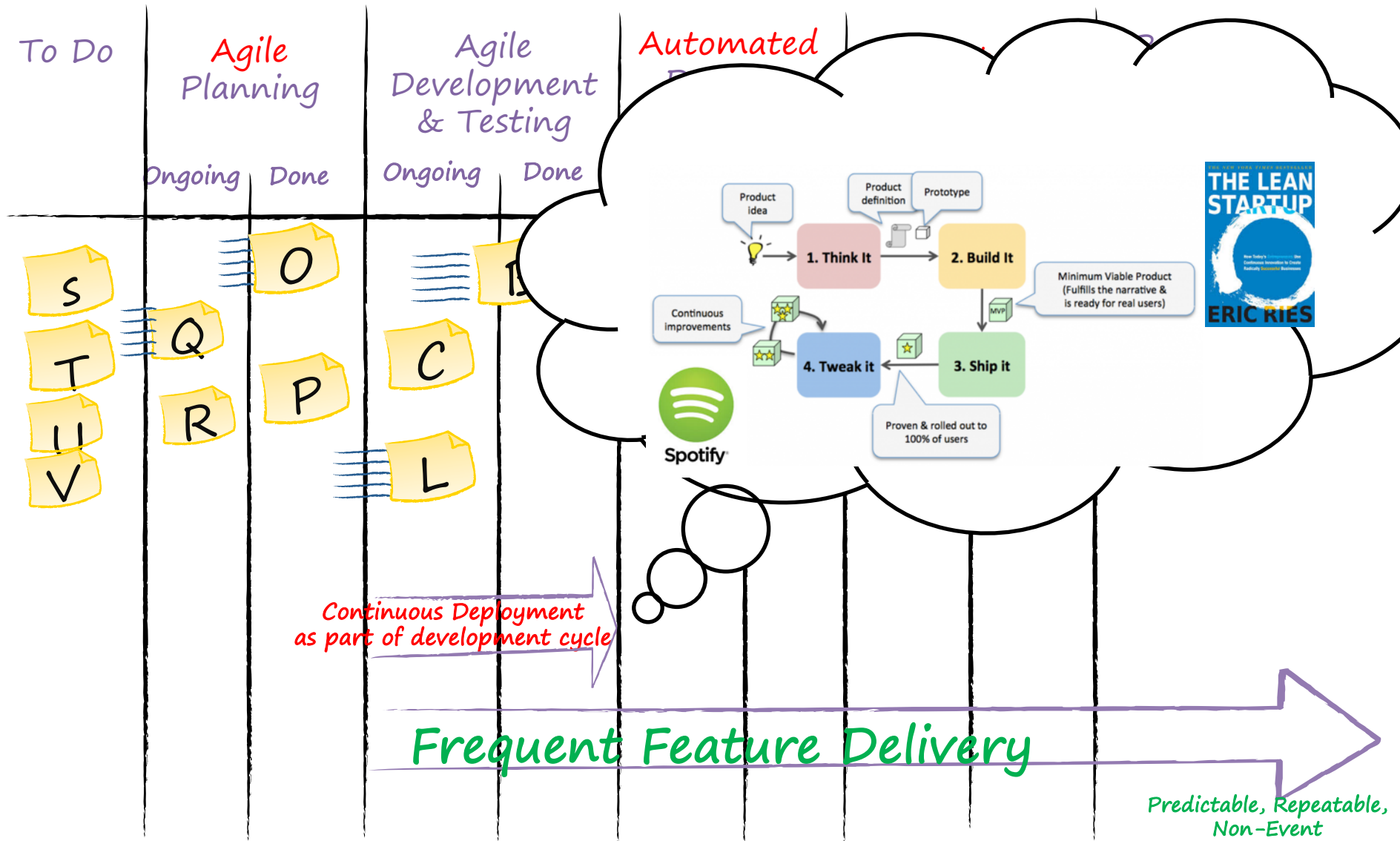
Agile Development



DevOps Vision

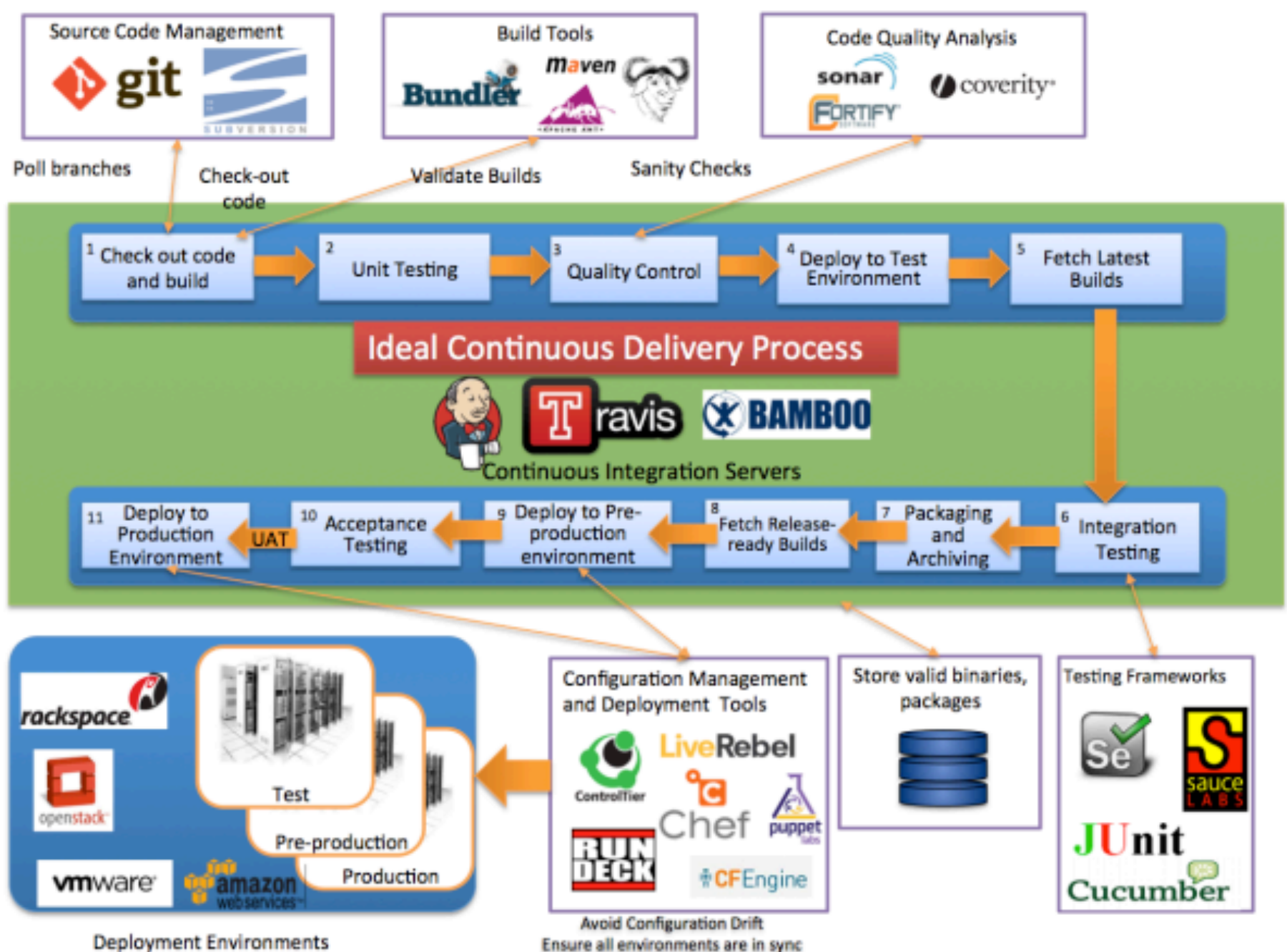


The real DevOps Vision



DevOps origins - WebOps



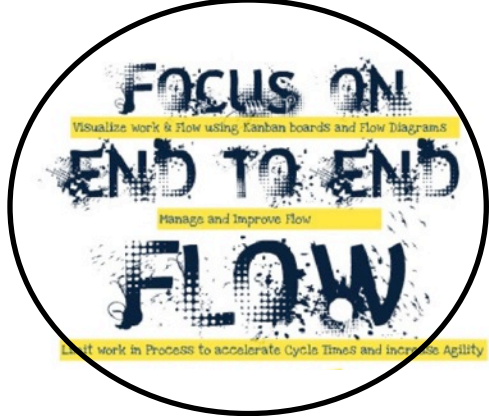


Enterprises are going DevOps as well...



But when the typical Enterprise meets DevOps...





KANBAN

A SANE WAY TOWARDS DEVOPS

RetroSpectives - Sense & respond to flow/feedback problems



FOCUS ON

Visualize work & Flow using Kanban boards and Flow Diagrams

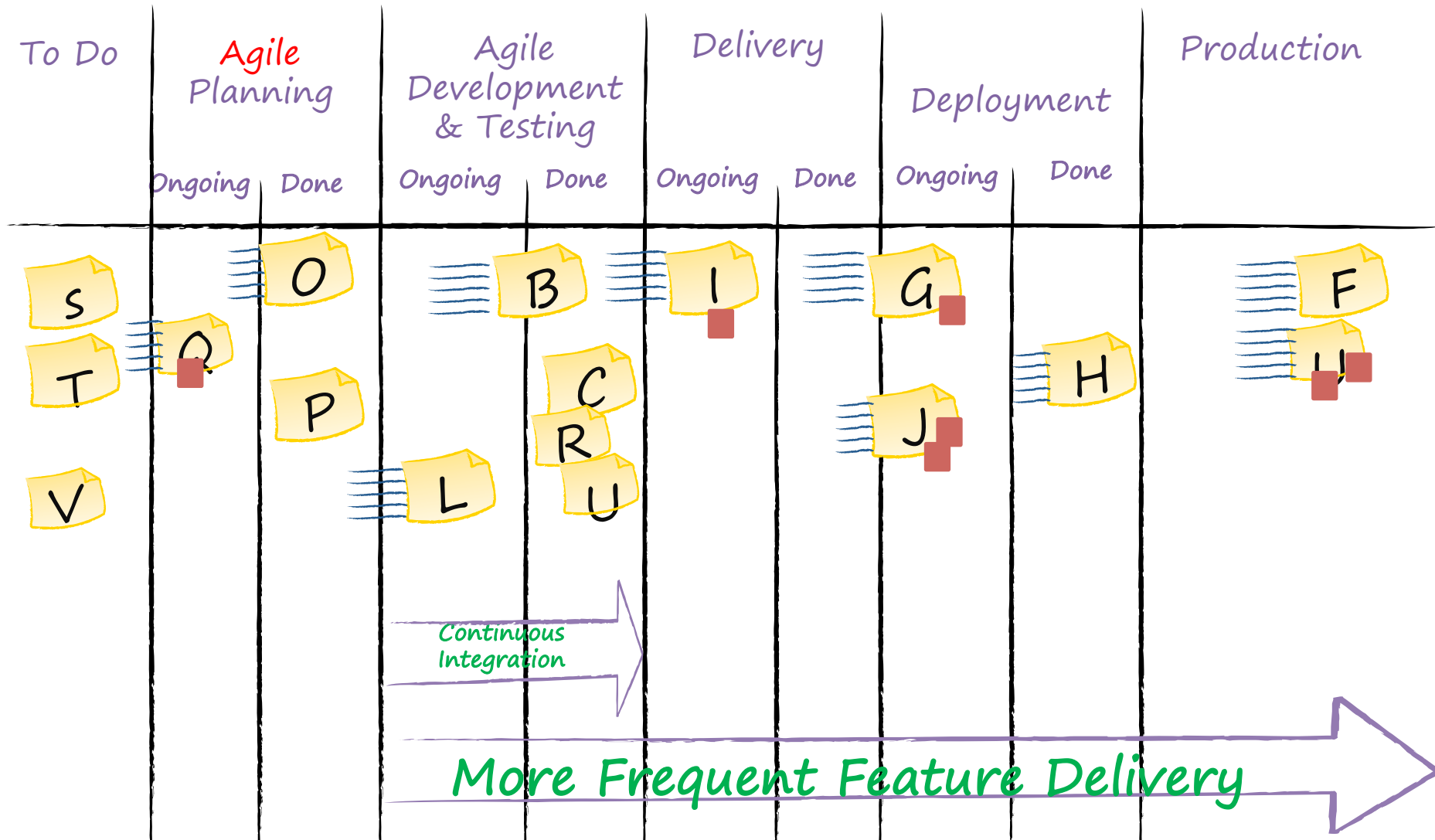
END TO END

Manage and Improve Flow

FLOW

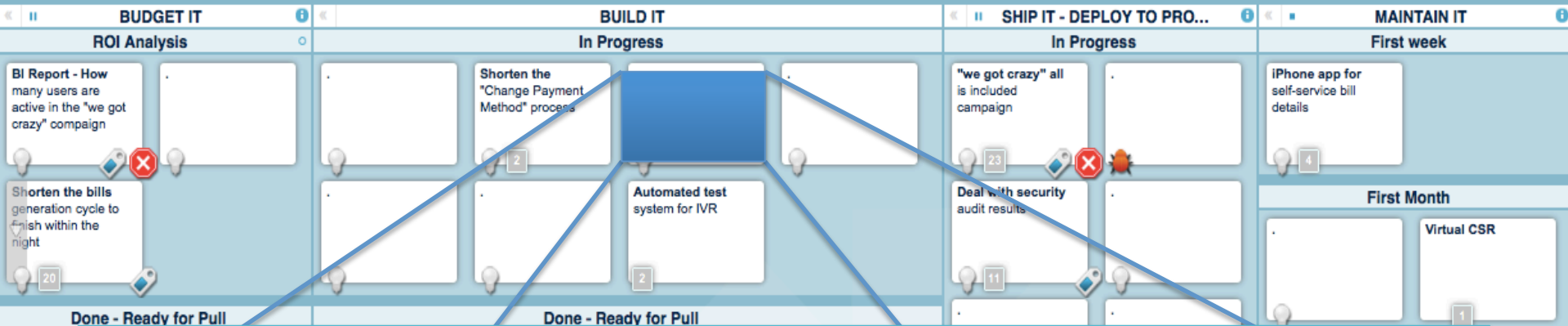
Limit work in Process to accelerate Cycle Times and increase Agility

Use Kanban to work towards Biz-Dev-Test-Ops Flow

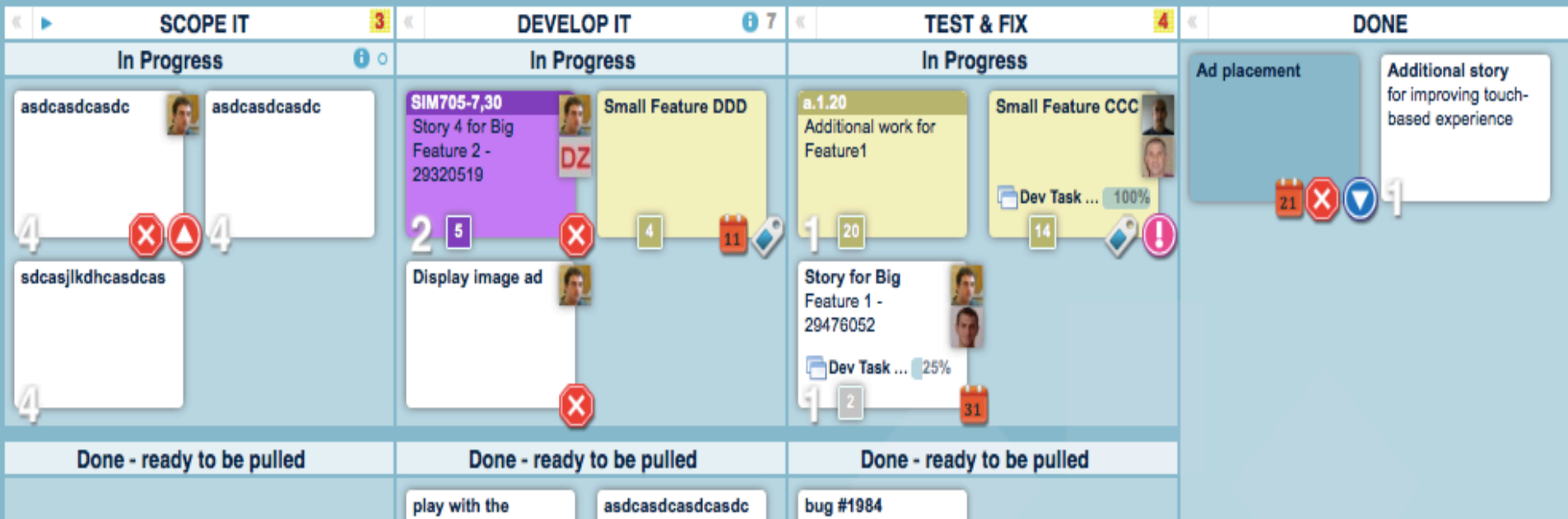


System Thinking - Visualize and improve flow across silos using Kanban boards or similar

Find your constraint/bottleneck and focus on it

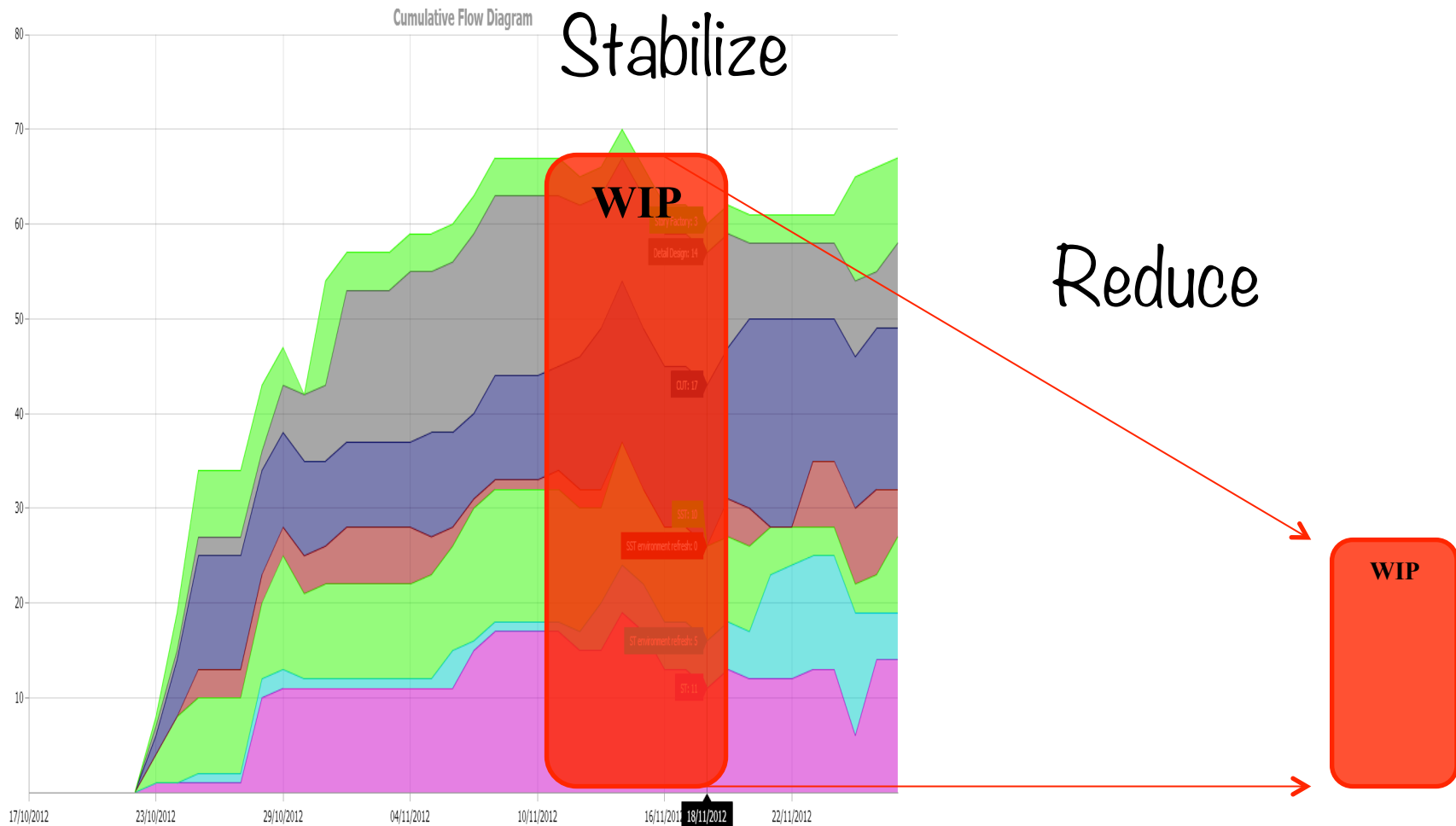


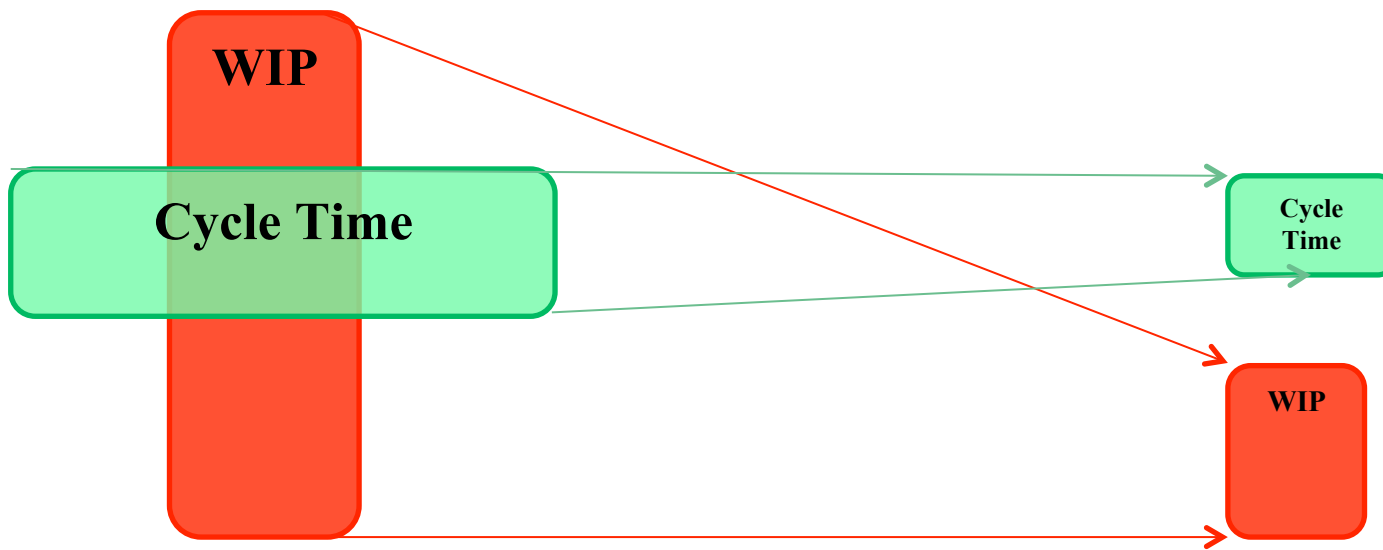
DevOps E2E board & flow



Agile Team board & flow

Key Measurement – Work in Progress –





Reduce WIP to create better flow and shorter cycle
Reduce at once/fast – Revolution
Reduce incrementally – Evolution

**FOCUS ON
END TO END
FLOW**

Visualize work & flow using Kanban boards and Flow Diagrams

Manage and Improve Flow

Limit work in Process to accelerate Cycle Times and increase Agility

Use smaller integrative valuable slices of work
**AMPLIFY
FEEDBACK
LOOPS**
Bring Dev • Test • Ops to work together all the way
Bring Ops feedback up front (Operations Driven Development)
Increase frequency of Delivery

KANBAN
**A SANE WAY TOWARDS
DEVOPS**

RetroSpectives - Sense & respond to flow/feedback problems

**CONTINUOUSLY
IMPROVE**

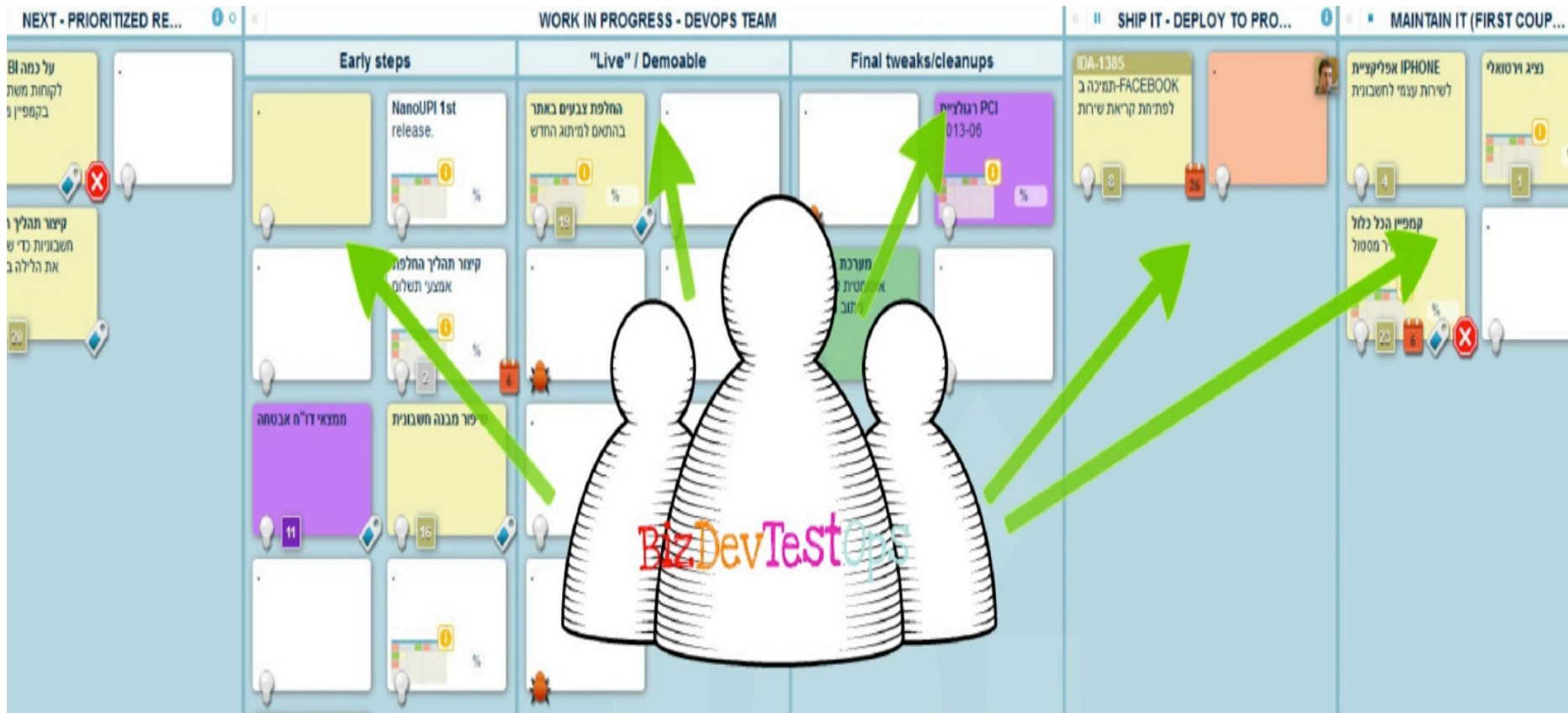
Reduce WIP / Accelerate loop to surface next opportunities for improvement

Assess Agile/DevOps practices/principles depth to suggest areas for improvement



Why wait? Bring feedback to the front!
Test/Ops Aware Development

Try Feature/Service/Product oriented BizDevTestOps Teams



**FOCUS ON
END TO END
FLOW**

Visualize work & flow using Kanban boards and Flow Diagrams

Manage and Improve Flow

Limit work in Process to accelerate Cycle Times and increase Agility

**AMPLIFY
FEEDBACK
LOOPS**

Use smaller integrative valuable slices of work

Bring Dev • Test • Ops to work together all the way

Bring Ops feedback up front (Operations Driven Development)

Increase frequency of Delivery

**KANBAN
A SANE WAY TOWARDS
DEVOPS**

RetroSpectives - Sense & respond to flow/feedback problems

**CONTINUOUSLY
IMPROVE**

Reduce WIP / Accelerate loop to surface next opportunities for improvement

Assess Agile/DevOps practices/principles depth to suggest areas for improvement

Leverage the friction/pain to drive/focus improvement efforts

Tighten the loop

Reduce Batch Size

Sense/Identify what hurts

e.g. deploy too labor-intensive

Implement reasonable fix for this speed

e.g. automate part of deploy

What might such a diet entail?

- More frequent reviews/approvals/integration/testing/deployment
- Changes to working teams
- Leaner processes with fewer/lighter gates
- Higher quality built in to avoid downstream blockages in Test/Deploy/Production
- Investment in automation/tooling in Dev/Ops to enable higher frequency
- Different metrics and score-cards used to govern the organization
- More DevOps-oriented organizational structure

Other steps are context-specific... You will have to find out!

High performing organizations:

Deploy code 30 times more frequently.

High performing organizations deploy code 30 times more often, and 8000 times faster than their peers, deploying multiple times a day, versus an average of once a month. Frequent deployments coupled with faster change lead times enable operational agility.

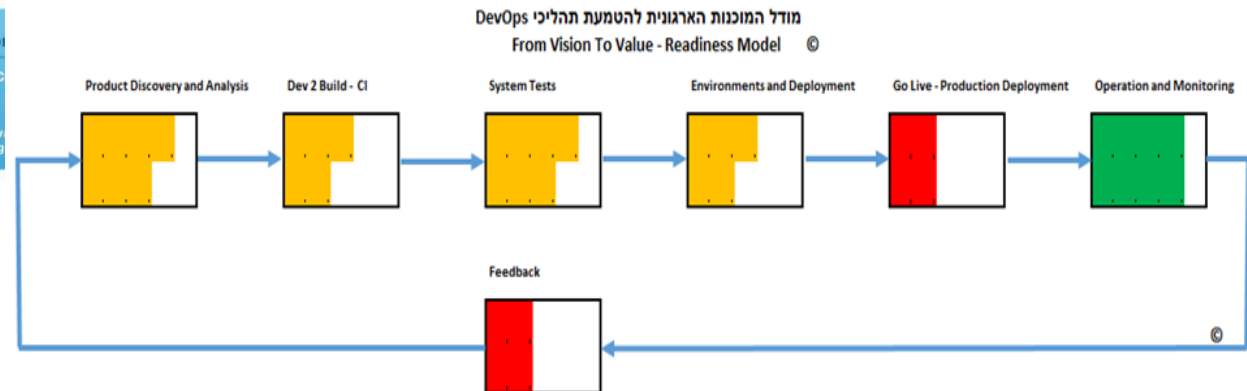
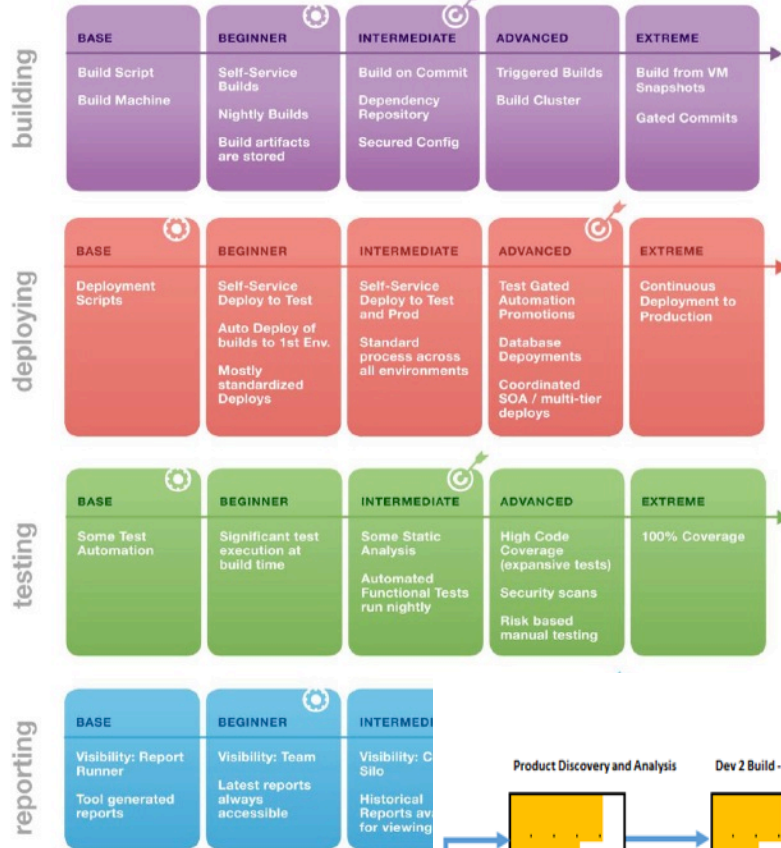
Have 50 percent fewer failures.

High performing organizations have double the change success rate and restore service 12 times faster than their peers. Fewer failures and faster recovery mean less risk to the business when changes are deployed.

DevOps == Continuous Deployment?

Assess your DevOps Maturity and take the next steps

<https://www.ibm.com/cloud/docs/continuous-delivery-maturity-model/>



*AgileSparks DevOps Readiness Model – Early Draft – Oded Tamir – ask him to show it to you today...

The First Way:
Systems Thinking

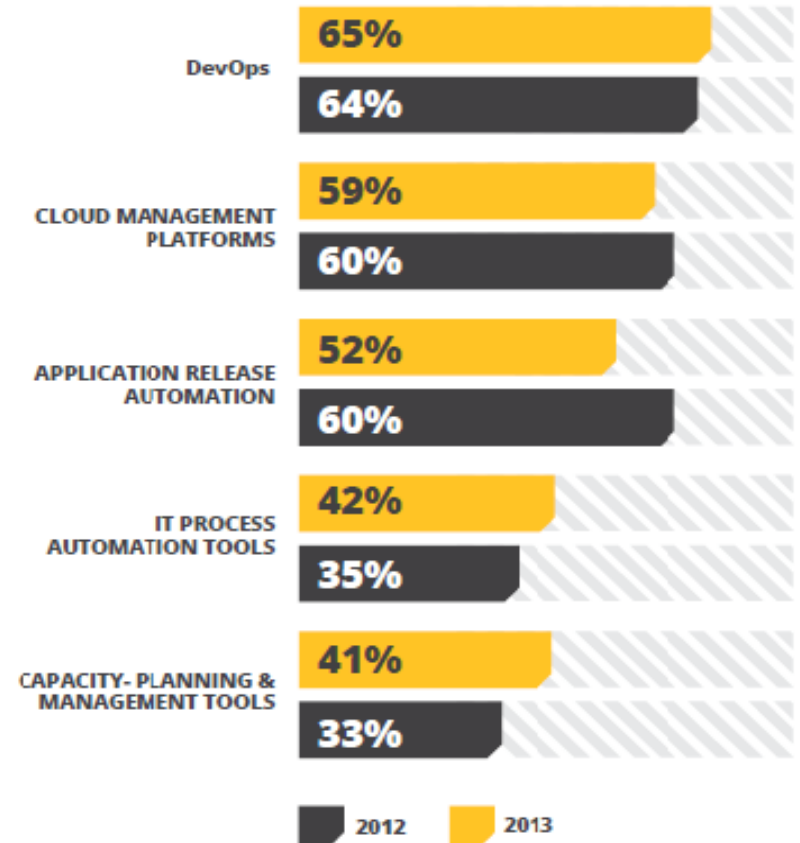
The diagram shows a curved arrow pointing from a box labeled 'Business' to a box labeled 'Customer'. The arrow is labeled 'Ops'.

```

graph LR
    Dev --> Ops
    Ops --> Dev
  
```



KEY INITIATIVES UNDER DISCUSSION

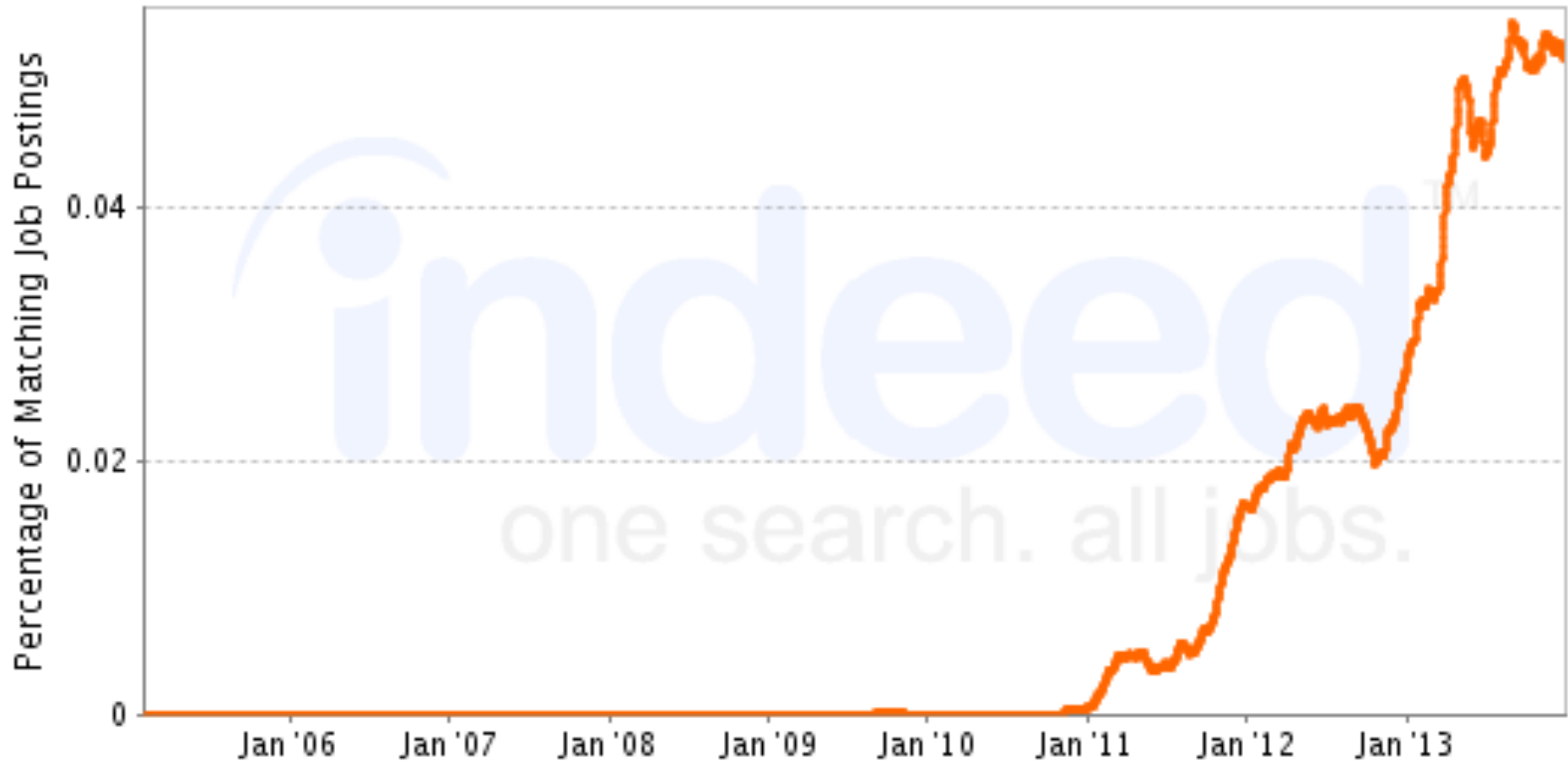


From a survey by Rebellabs - A total of 64% of respondents said that DevOps was a key initiative within their organization in 2012. This figure stepped up to 65% in 2013:

DevOps positions on the rise...

Job Trends from Indeed.com

— DevOps



The Journey towards DevOps

- Whether you are a startup, an established enterprise software company or an IT unit getting into the Cloud/SaaS world you should start to pay attention to DevOps
- Our DevOps management workshops can help you understand how it can help you, what style of DevOps is a good fit for your context, and plan your journey.
- Look forward to more DevOps services from AgileSparks later this year – inform us at www.agilesparks.com or info@agilesparks.com to be the first to know.

