

Automation and Eliminating Toil



Elton Stoneman
Freelance Consultant and Trainer
@EltonStoneman | blog.sixeyed.com



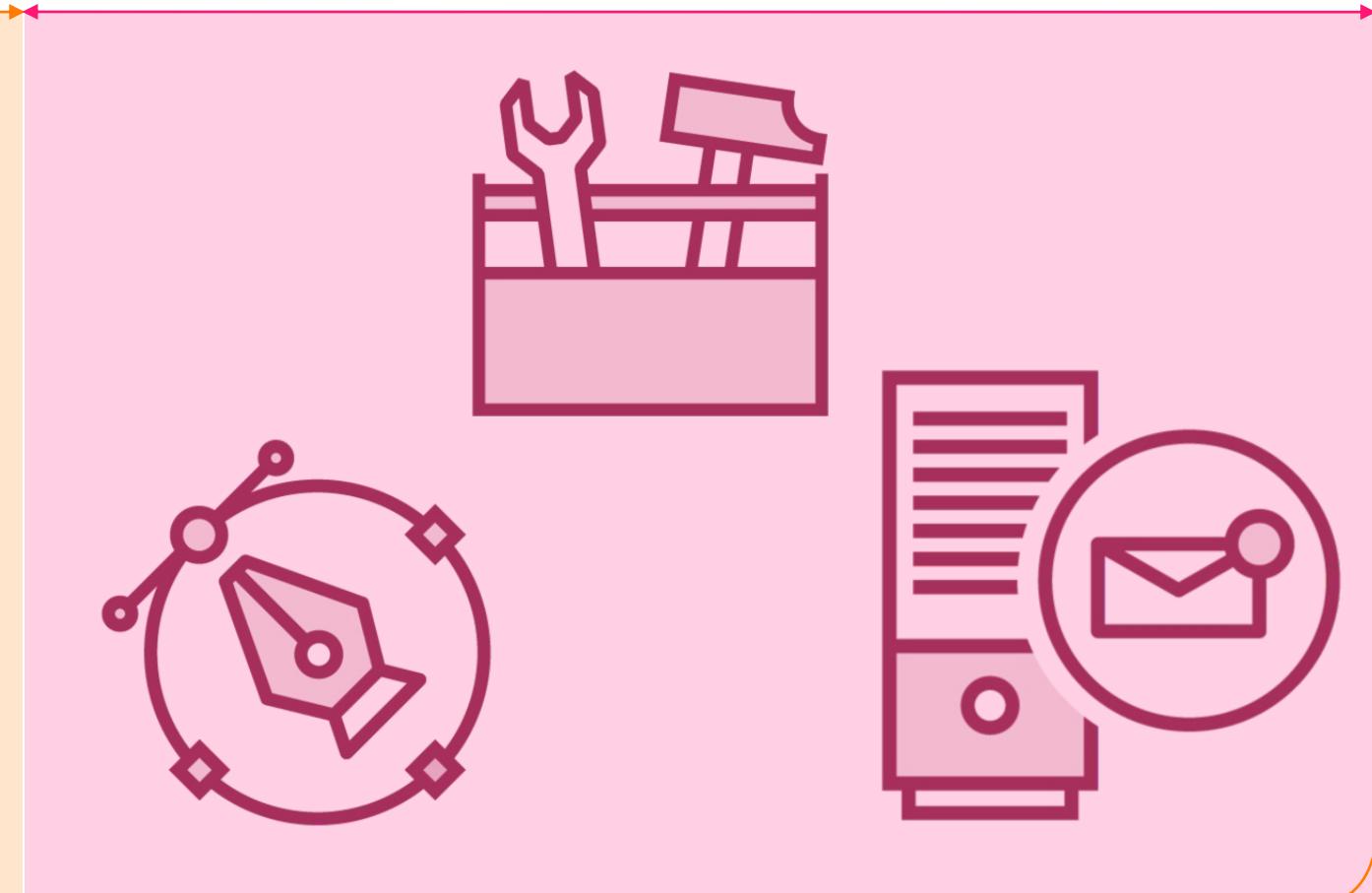
Overhead



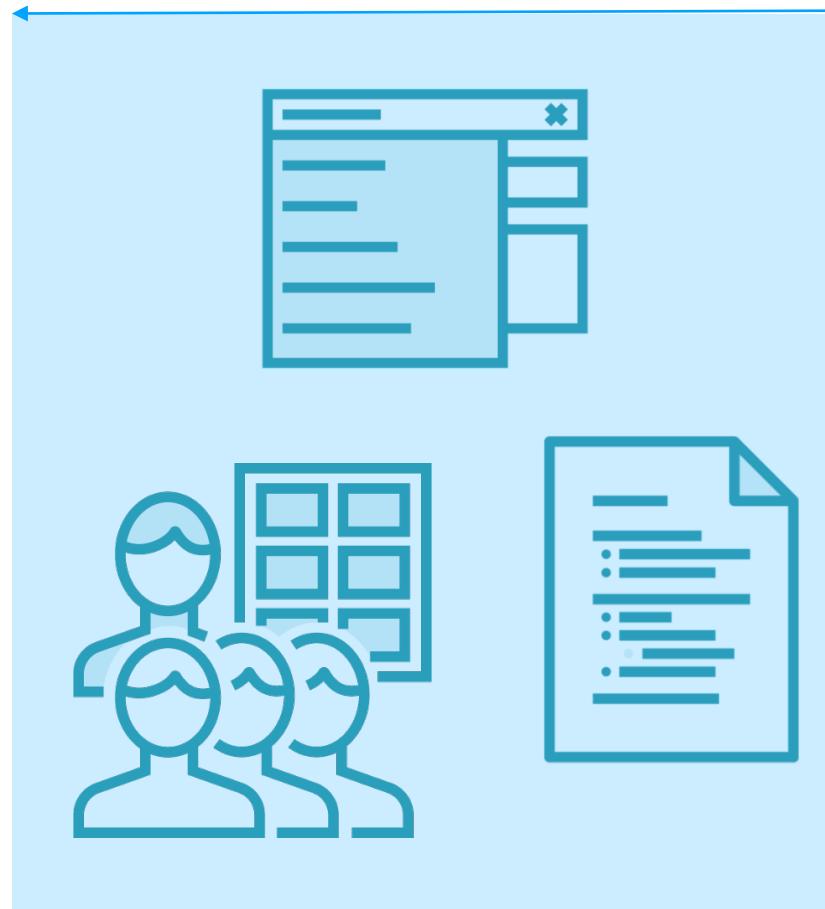
Toil



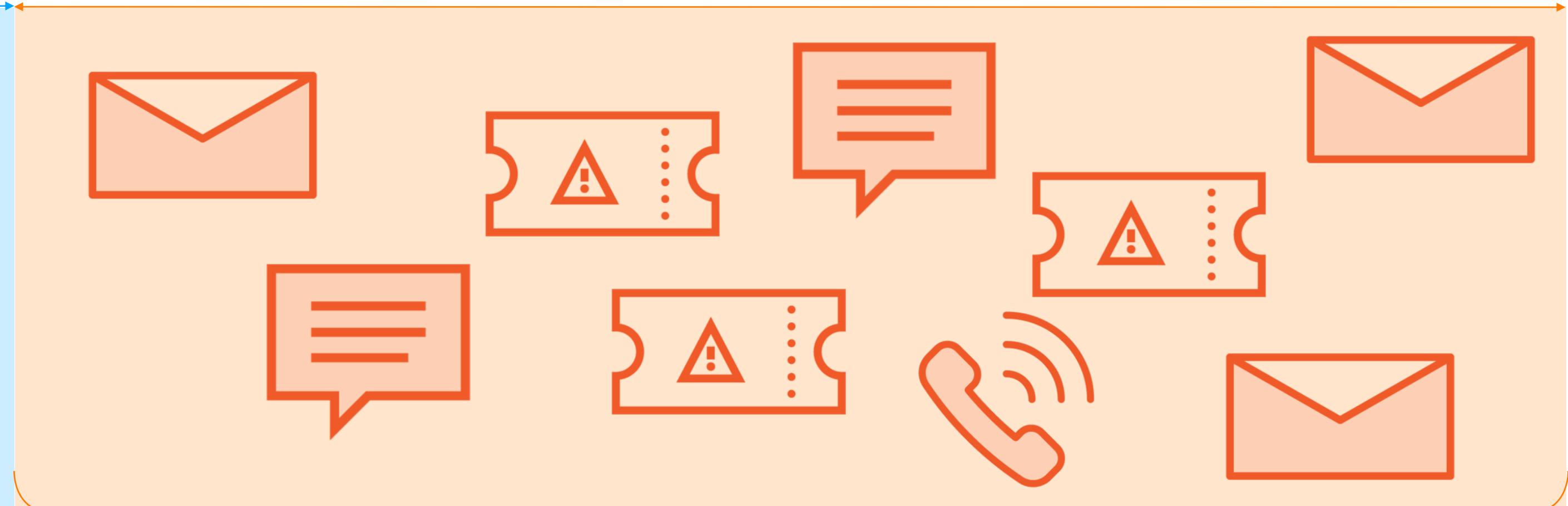
Strategic



Overhead



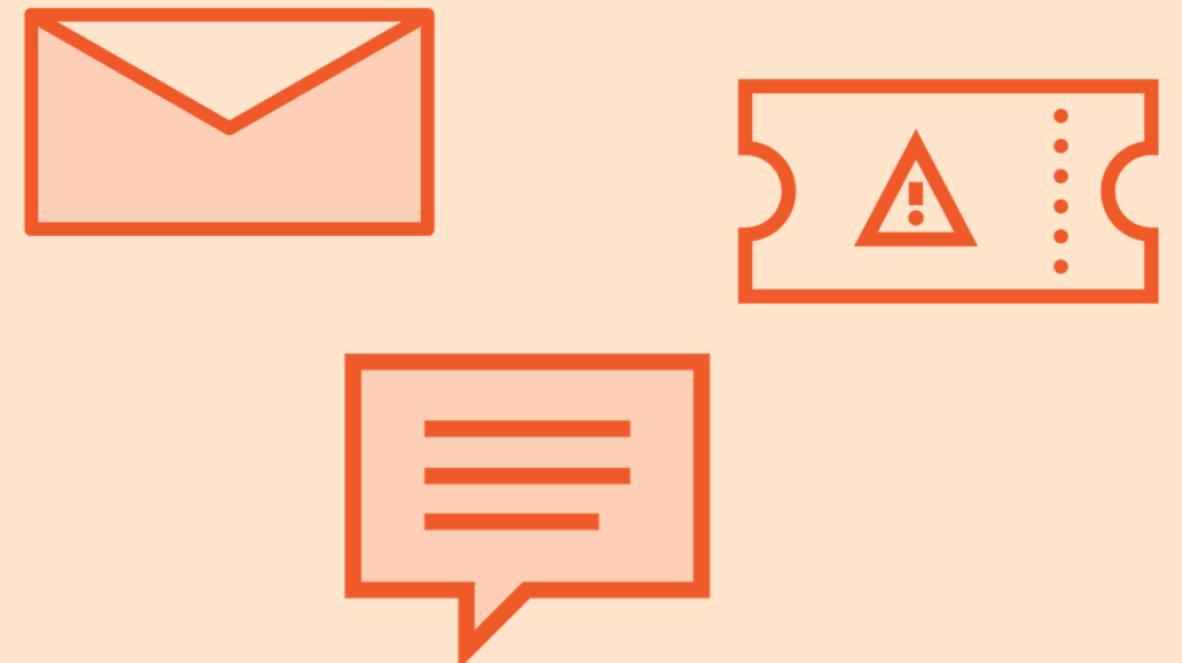
Toil



Overhead



Toil : max. 50%



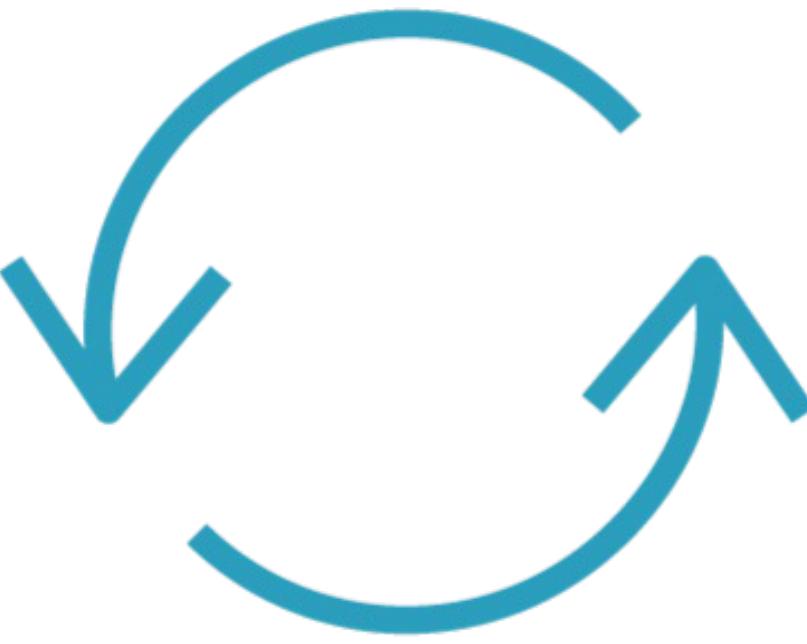
Strategic



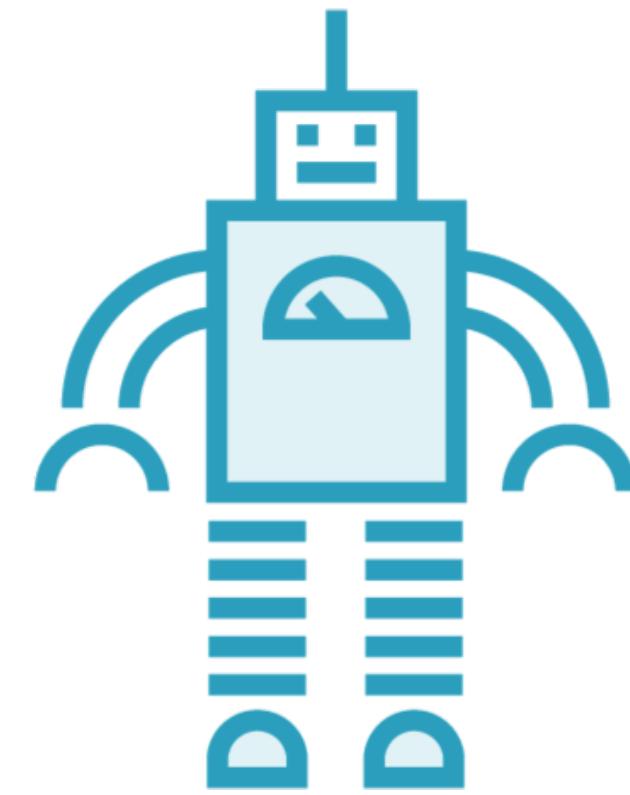
What is Toil?



Labor Intensive



Repetitive



Automatable





App not responding

Raise ticket



Log in to server

Restart process

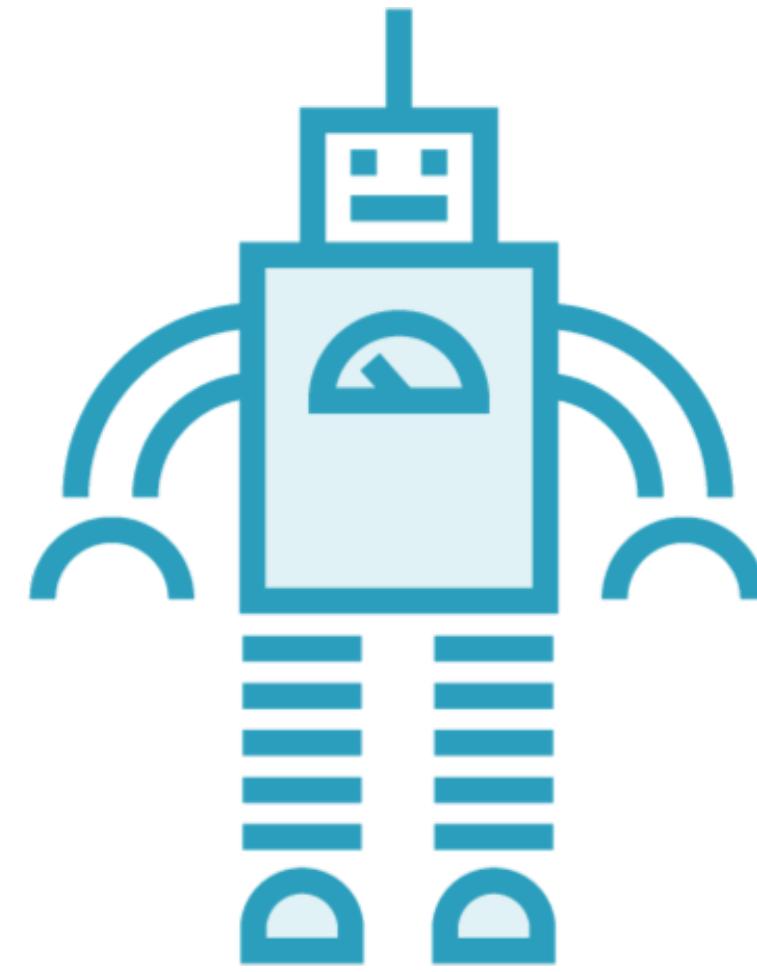
- **Manual**
- **Repetitive**
- **Automatable**
- **Reactive**
- **Low-value**
- **Scales linearly**



Is it Toil?



"I can do this!"



"Me too! Bloop."



Engineering in SRE



Strategic



High Value



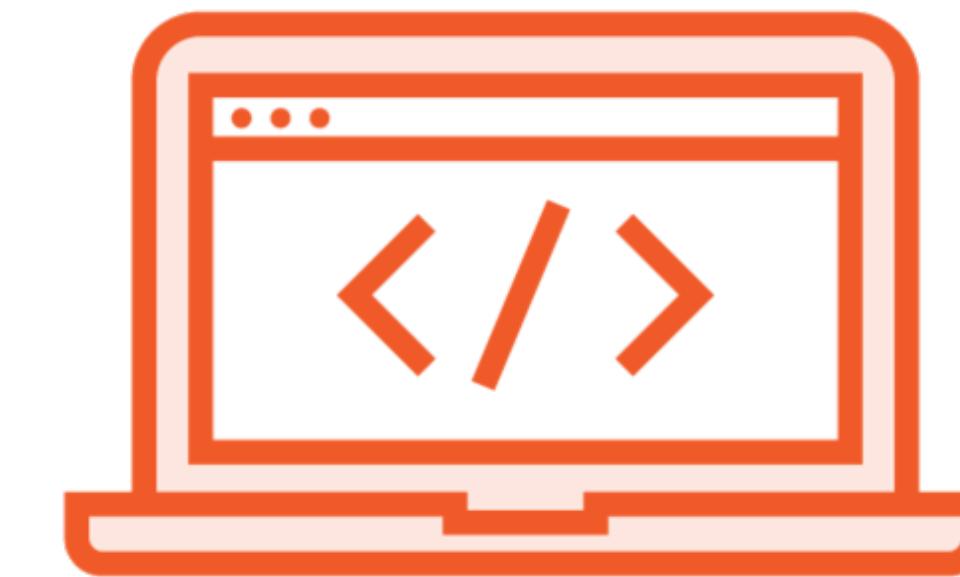
Human



Engineering in SRE

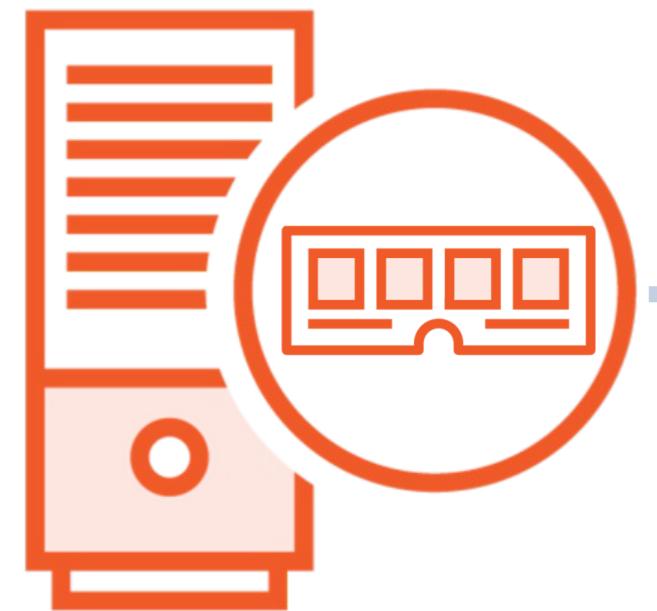


Systems Engineering
Configuration
Tuning



Software Engineering
Reliability
Scalability



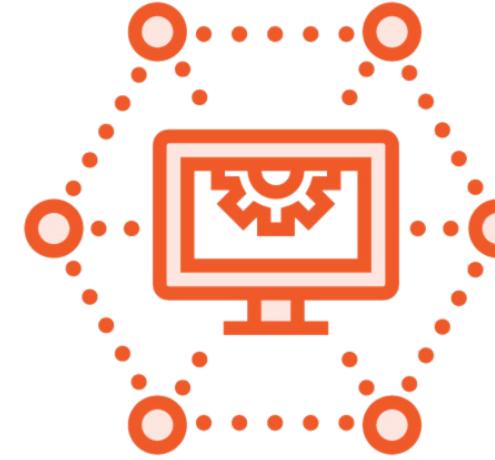


App not responding

Raise ticket

Log in to server

Restart process

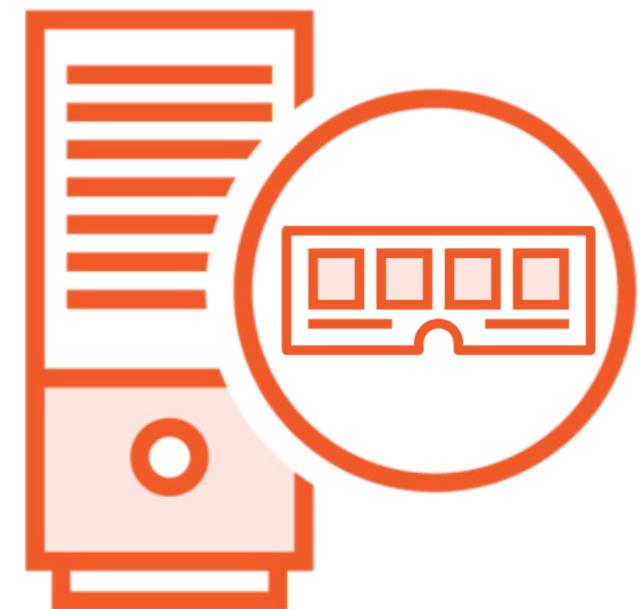


- **Runtime change**
- **Use platform capabilities**
- **Doesn't fix the issue**



Move to container with healthcheck

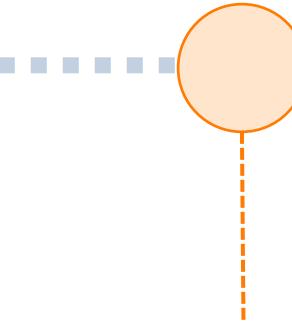




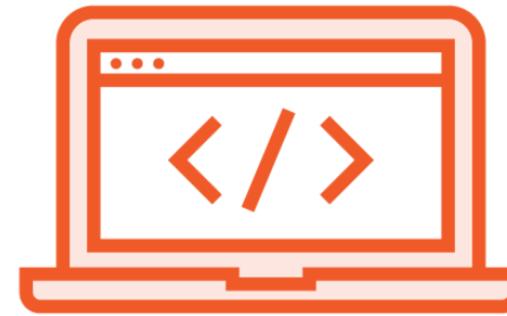
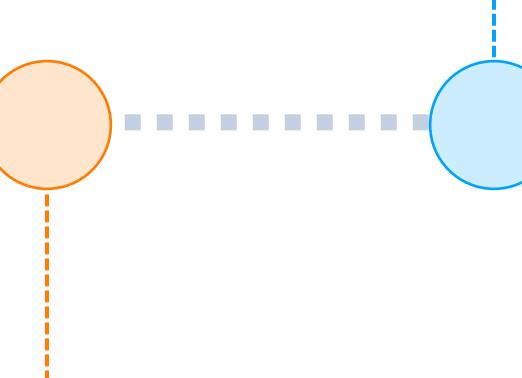
App not responding

Raise ticket

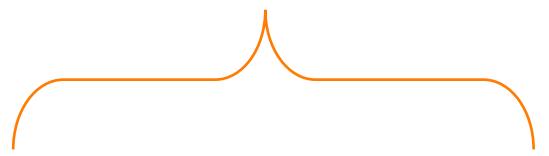
Log in to server



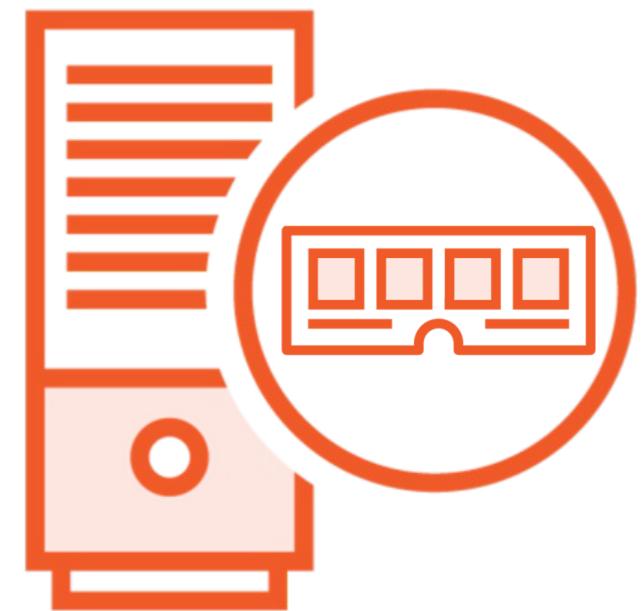
Restart process



- **Permanent fix**
- **More work**
- **Introduces risk**



Fix leaks



App not responding

Raise ticket

Log in to server

Move to container with healthcheck

Restart process

Fix leaks





Restricting Toil to 50%



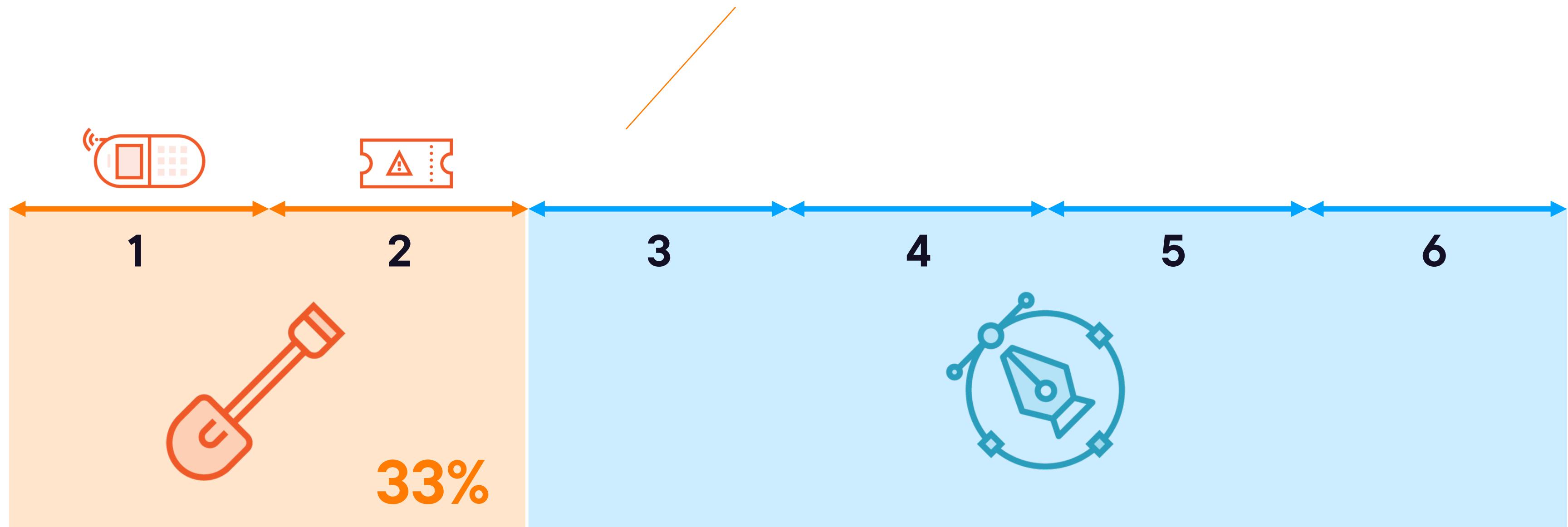
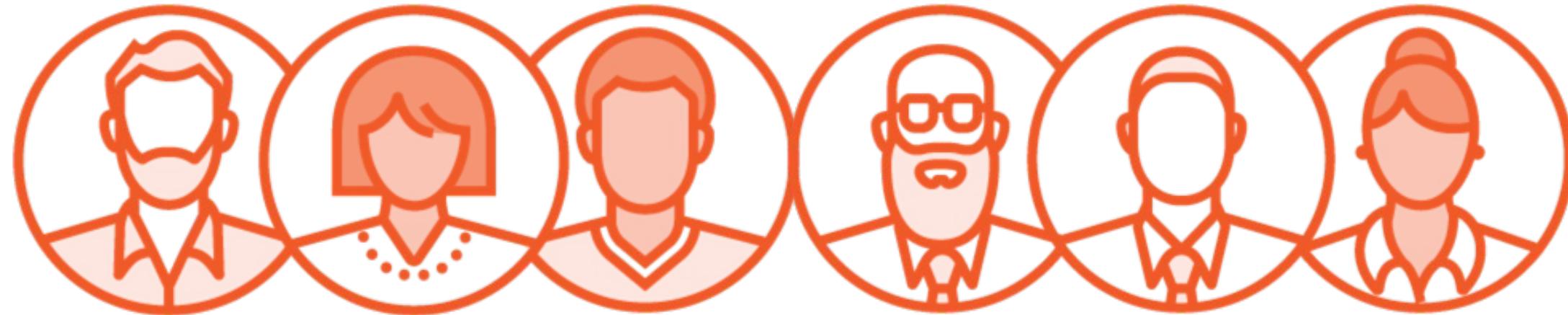
Stated limit

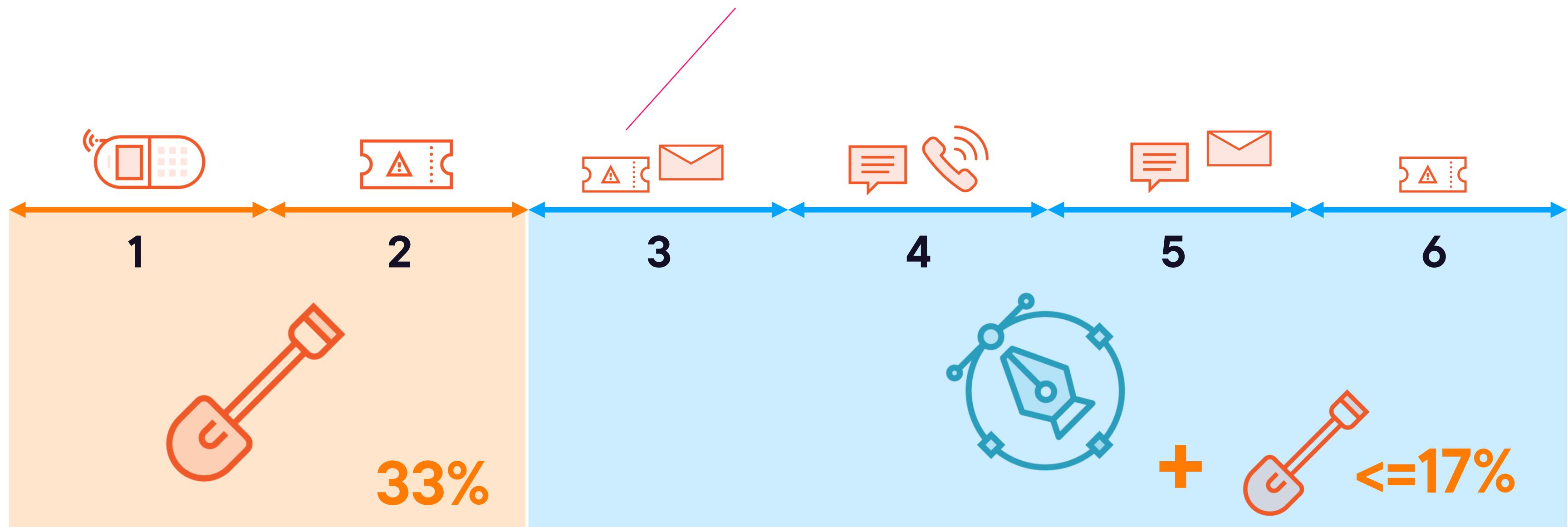
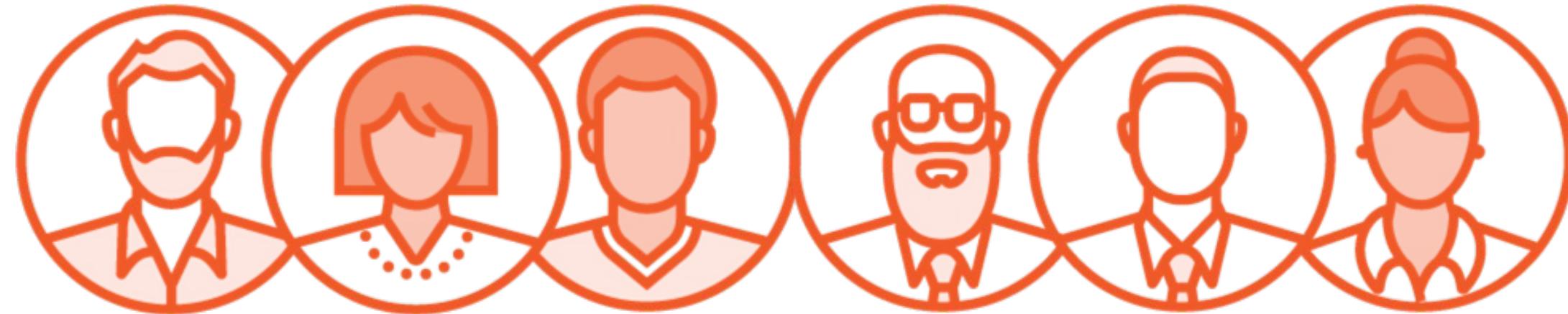
- Google use 50%
- Publicly communicated
- Commitment to the team

Toil guarantee

- Part of the job spec
- Management support
- Clear difference from ops









Toil can be attractive

- Quick fixes
- Sense of achievement
- But it's short-term

And it's bad for morale

- Team's energy dips
- Strategic projects stall
- People move on



Identifying and Measuring Toil





Data-driven analysis

- Identify toil
- Quantify ongoing costs

Toil-reduction backlog

- Prioritize projects
- SREs or whole team progress

Cost-benefit analysis

- Objective measures of toil
- Effort or elapsed time





Add to Azure
subscription

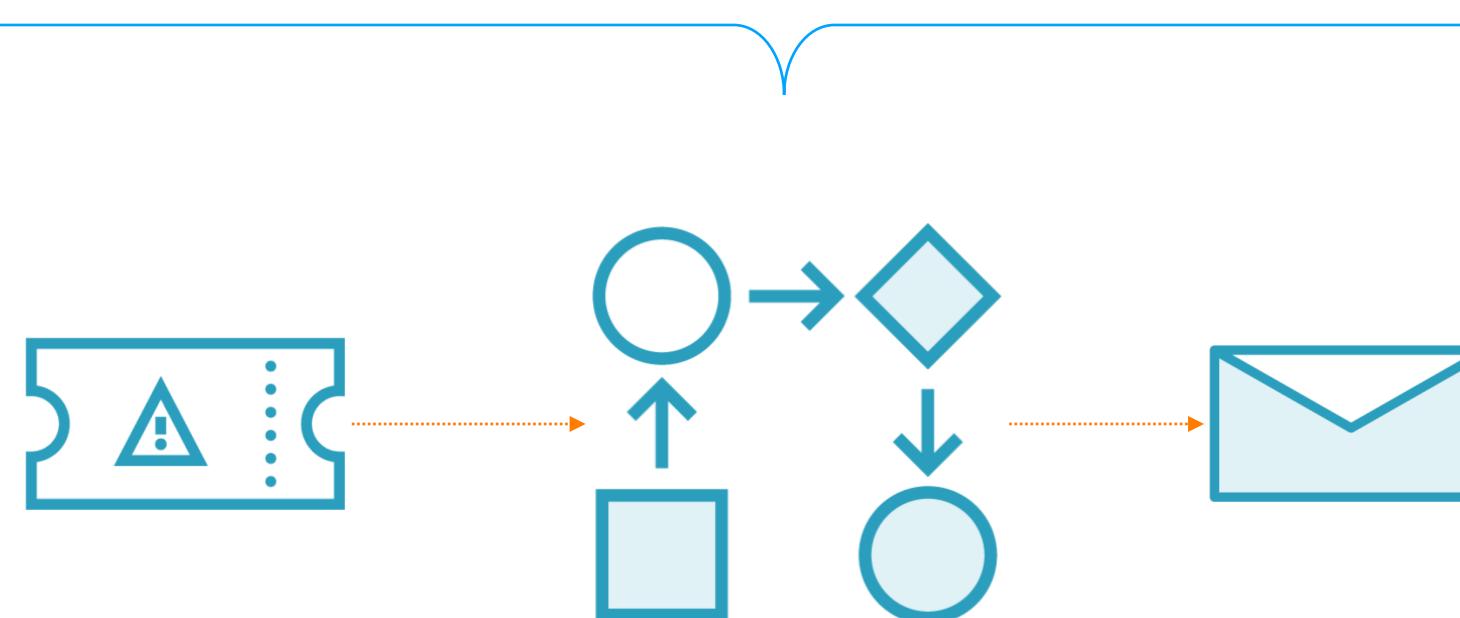
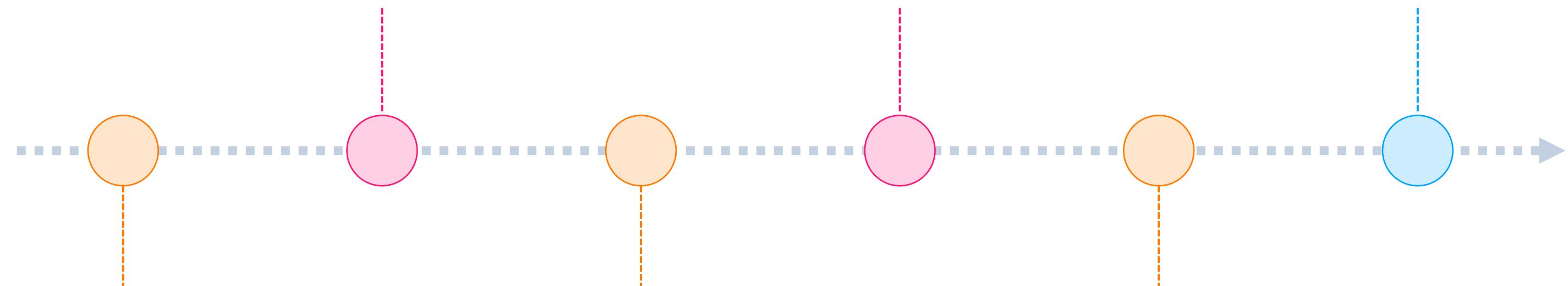
Add to system X
as read-only

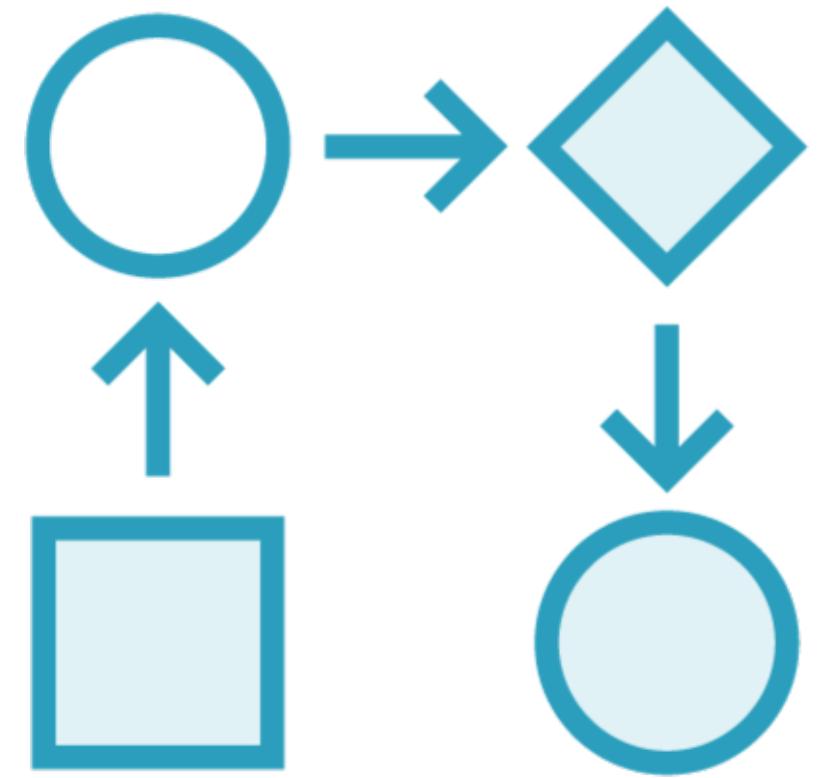
Automate?

Add to GitHub
repos

Add to Slack
channels

Add to system Y
as contributor





Automation projects

- Software engineering
- Support, maintenance, updates

Build costs vs. toil cost

- Time to value
- Maintenance and support
- Sharing with other teams

Other costs: \$

- Server outages
- Unused capacity

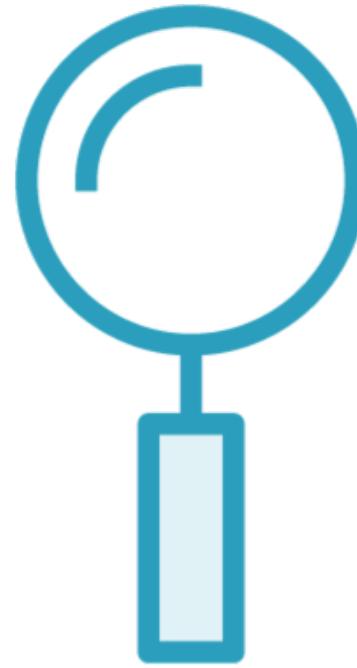




Engineering Away Toil



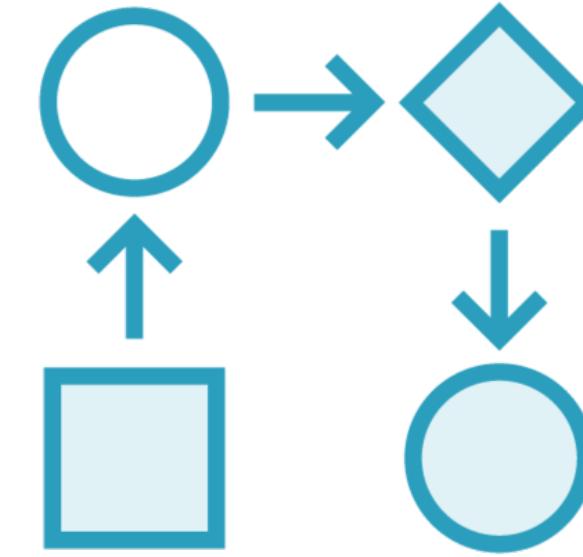
Identifying Automation Candidates



Analysis

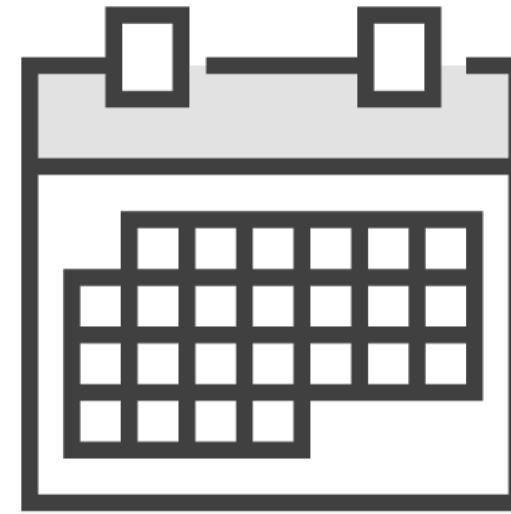


Documentation

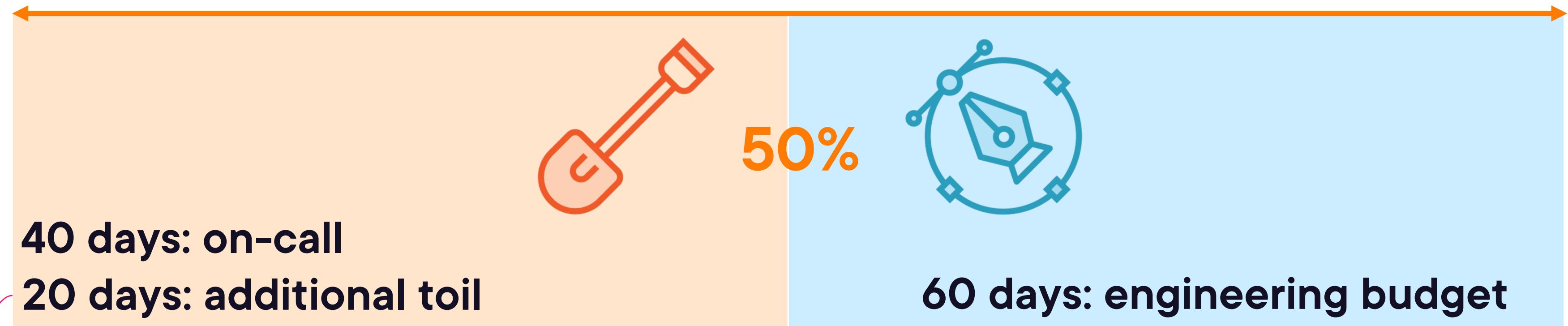


Repeatable Process





20 working days x6 SREs = 120 days



Top Toil Targets

		<i>Effort</i>	<i>Frequency</i>	<i>Days /month</i>
#1	 On-boarding	1 day	x5	5
#2	 Patching	2 days	x2	4
#3	 Provisioning	1 day	x3	3





Add to Azure
subscription

Add to GitHub
repos

No API!

Add to system X
as read-only

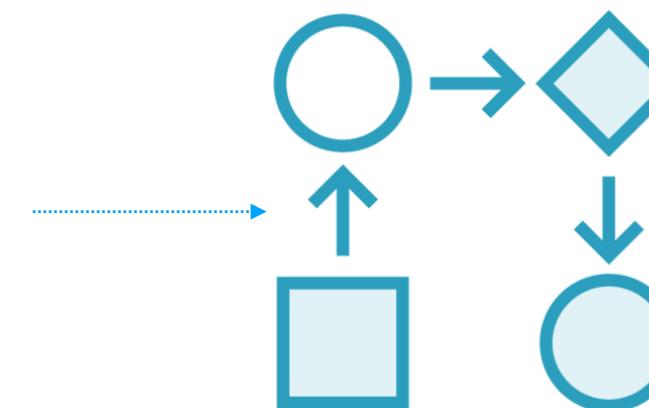
Automate?

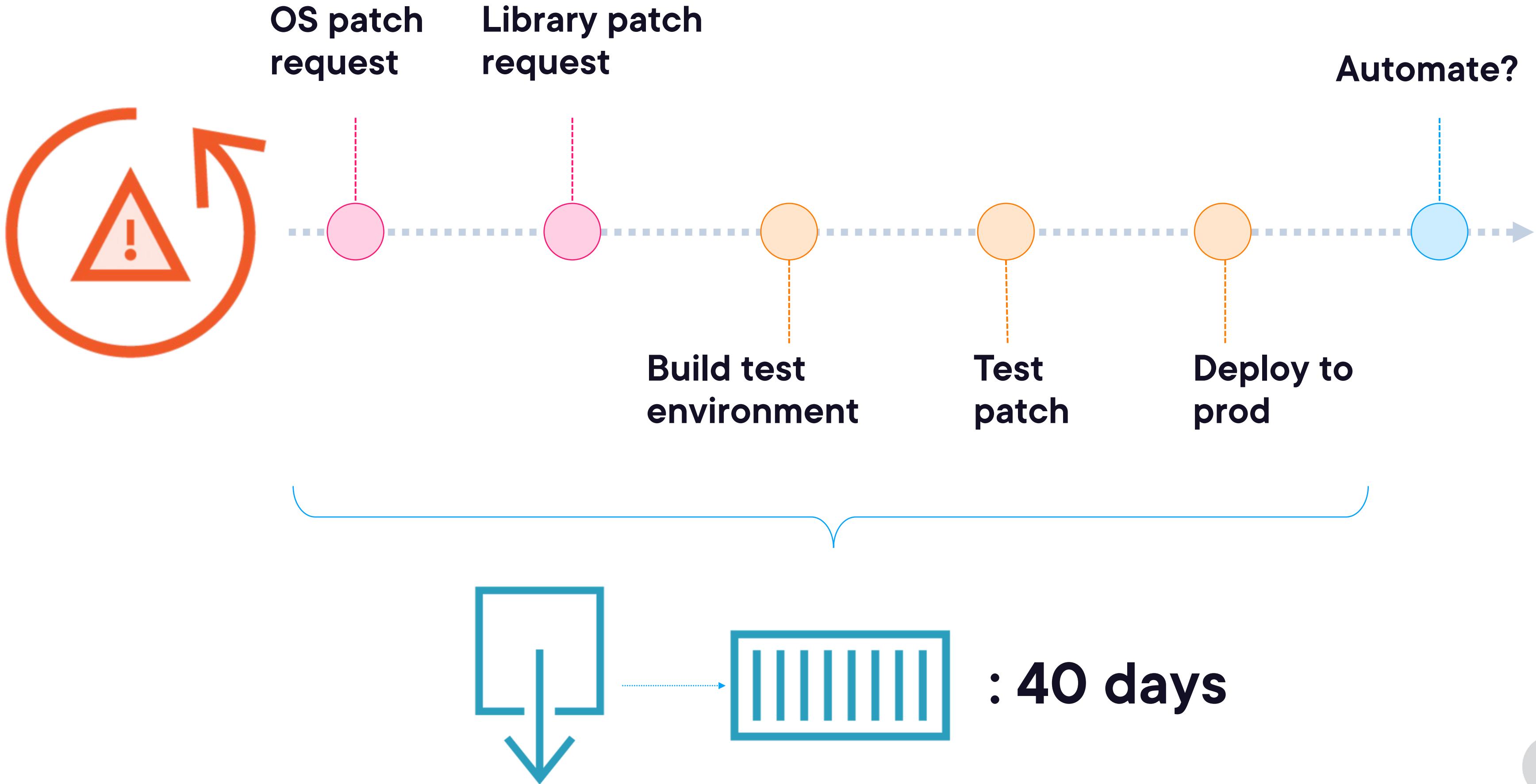
Add to Slack
channels

Add to system Y
as contributor

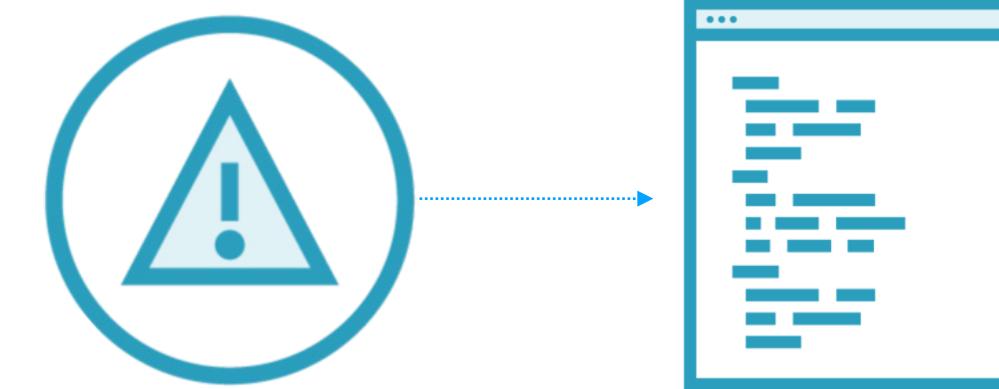
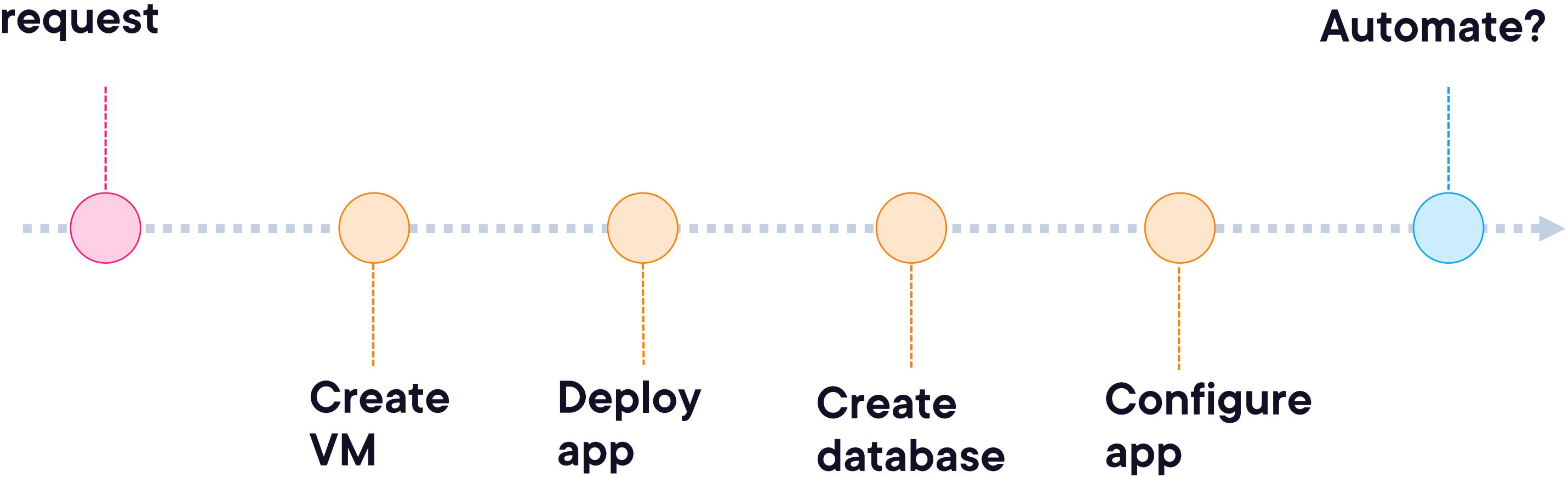
No API!!

: 65 days



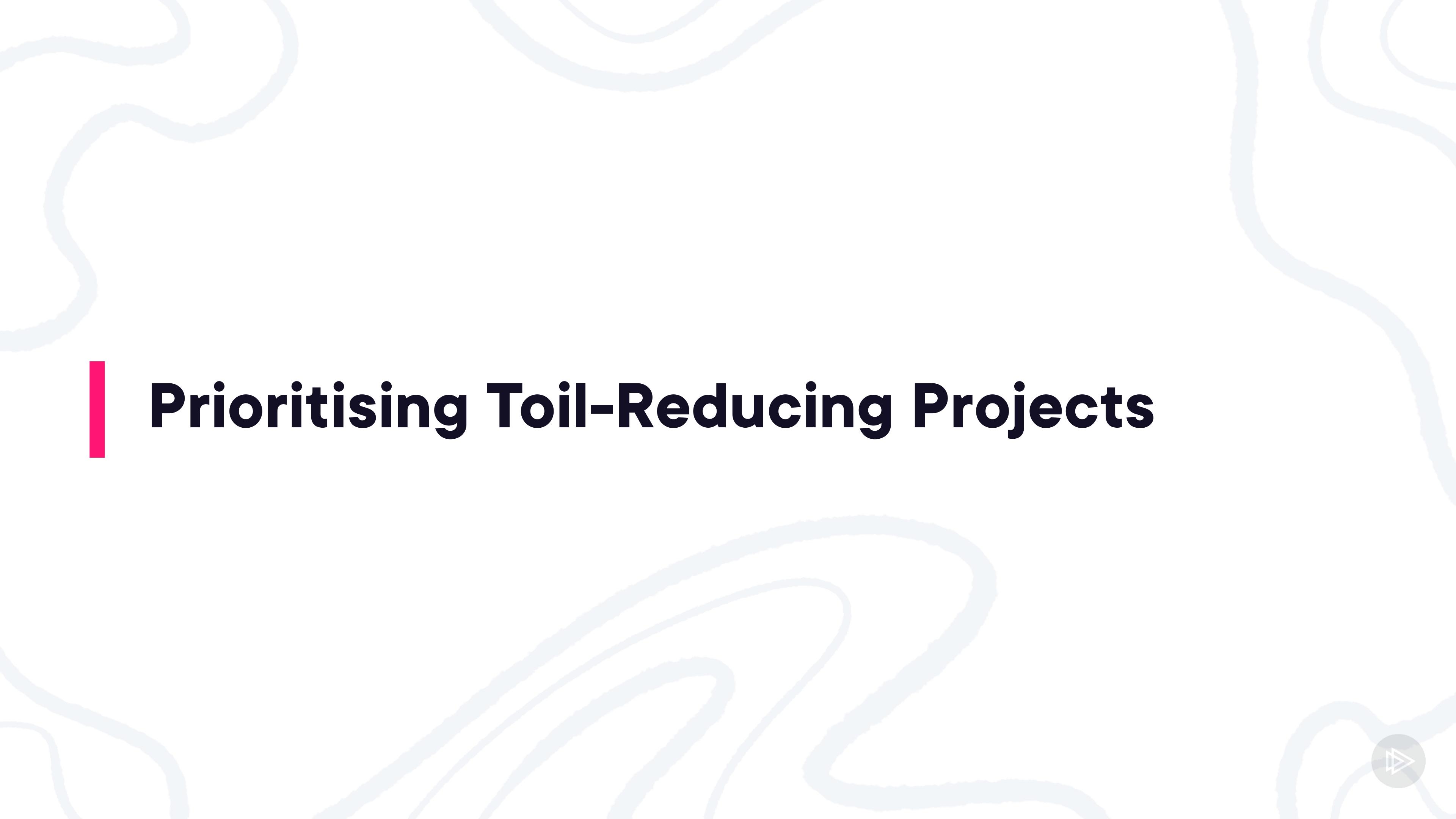


**New env
request**



: 10 days

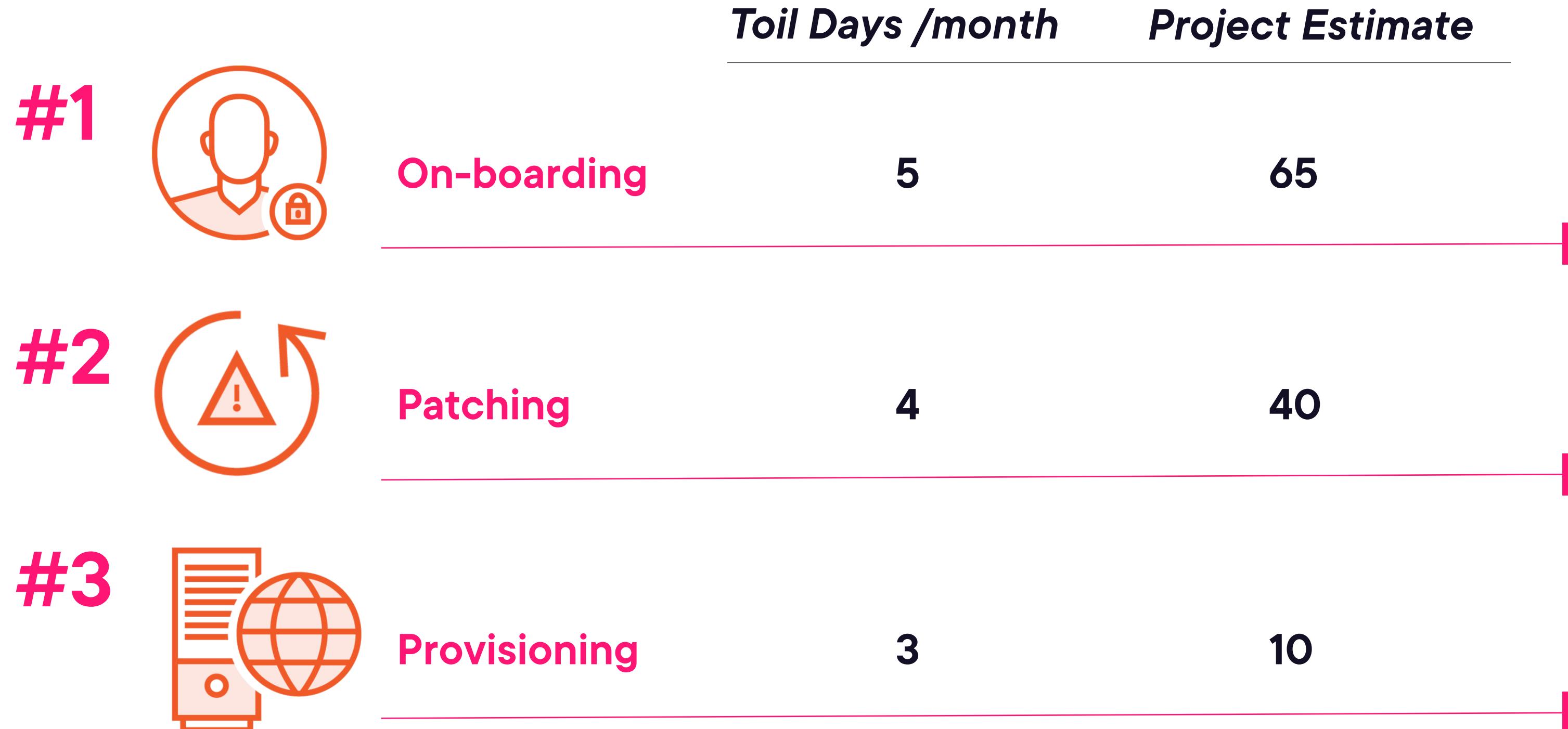




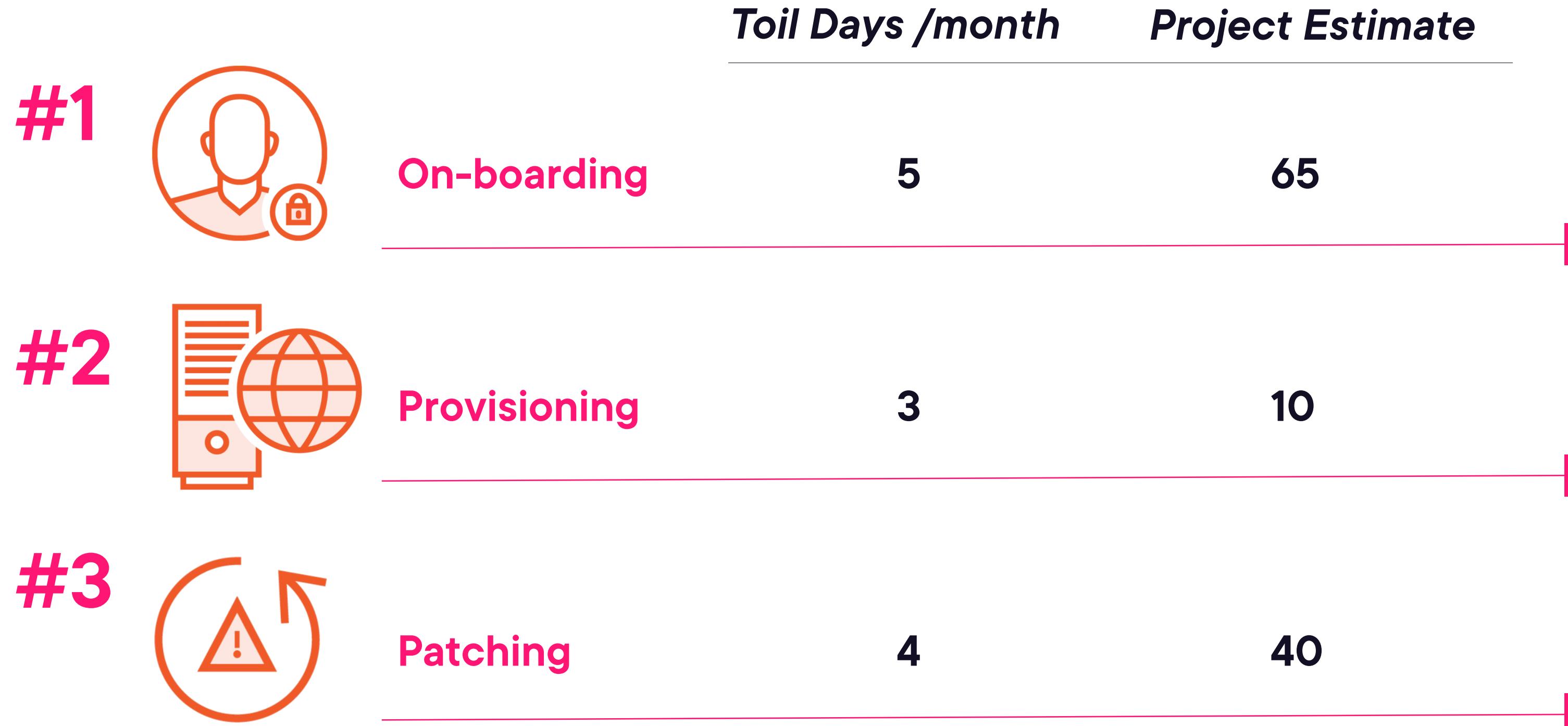
Prioritising Toil-Reducing Projects

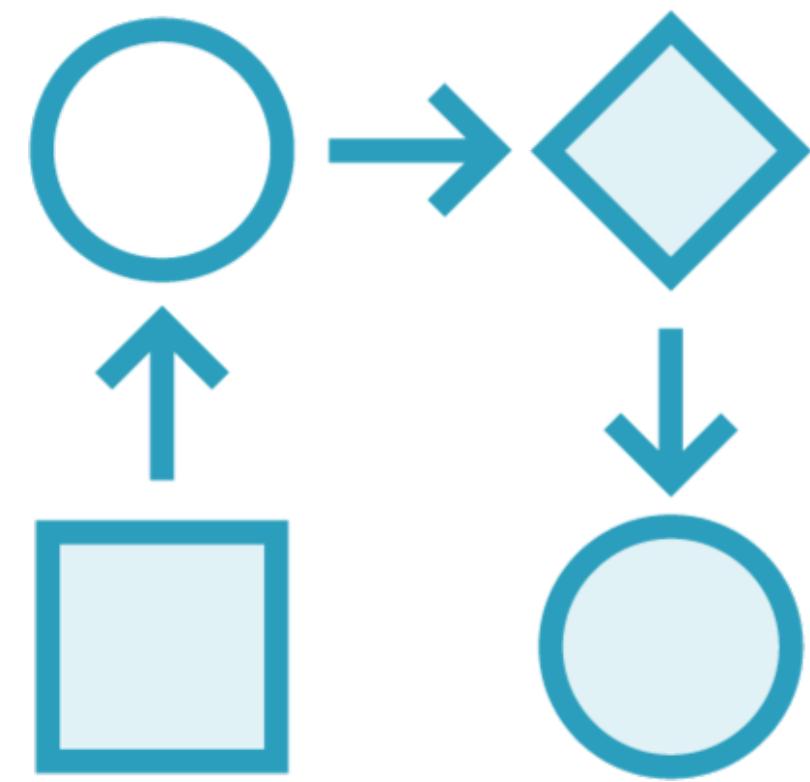


Top Toil Projects



Top Toil Projects





Toil-reduction side effects

- Productivity increase
- System reliability and availability
- Toolset standardization
- System simplification
- Culture of automation



#1?



Add to Azure
subscription

Add to system X
as read-only

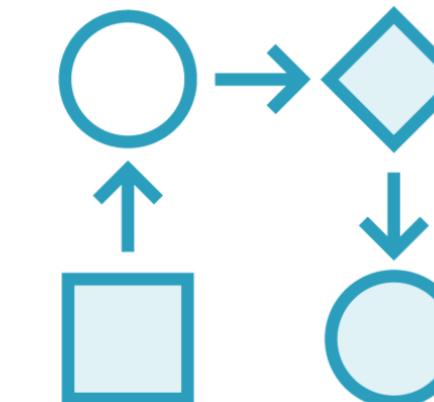
Automate?

Add to GitHub
repos

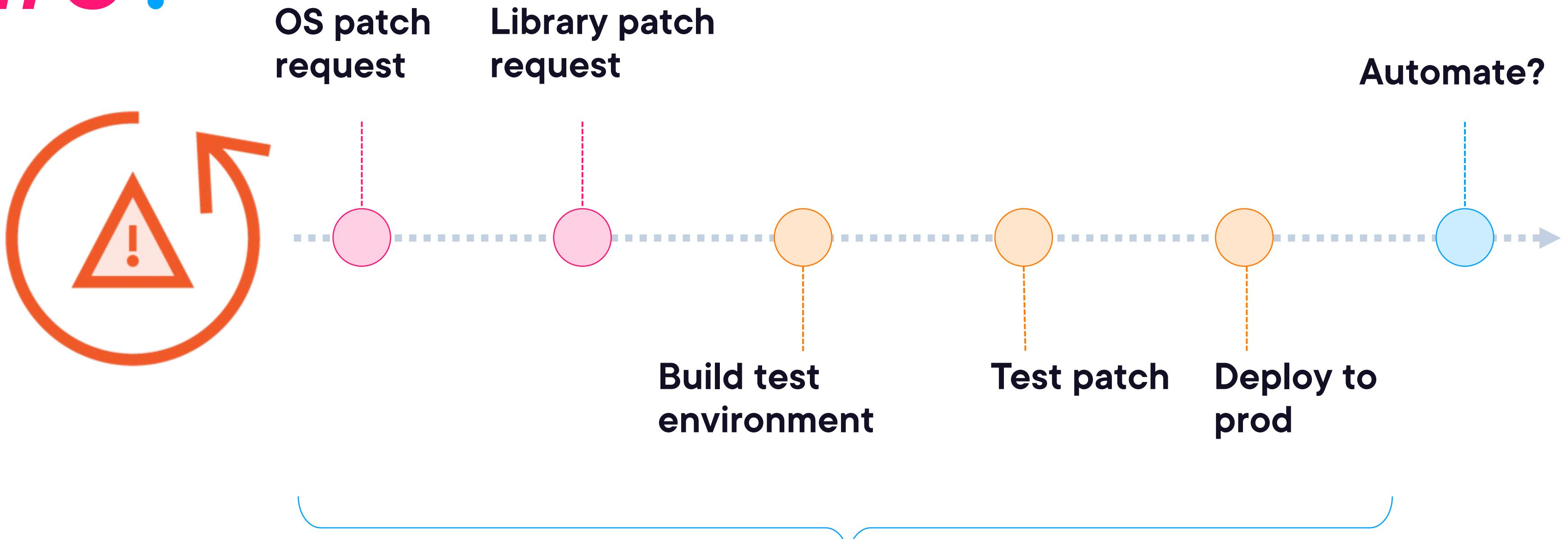
Add to Slack
channels

Add to system Y
as contributor

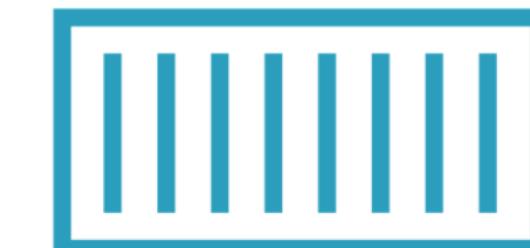
- Substantial software engineering
- Limited re-use
- No other side effects



#3?



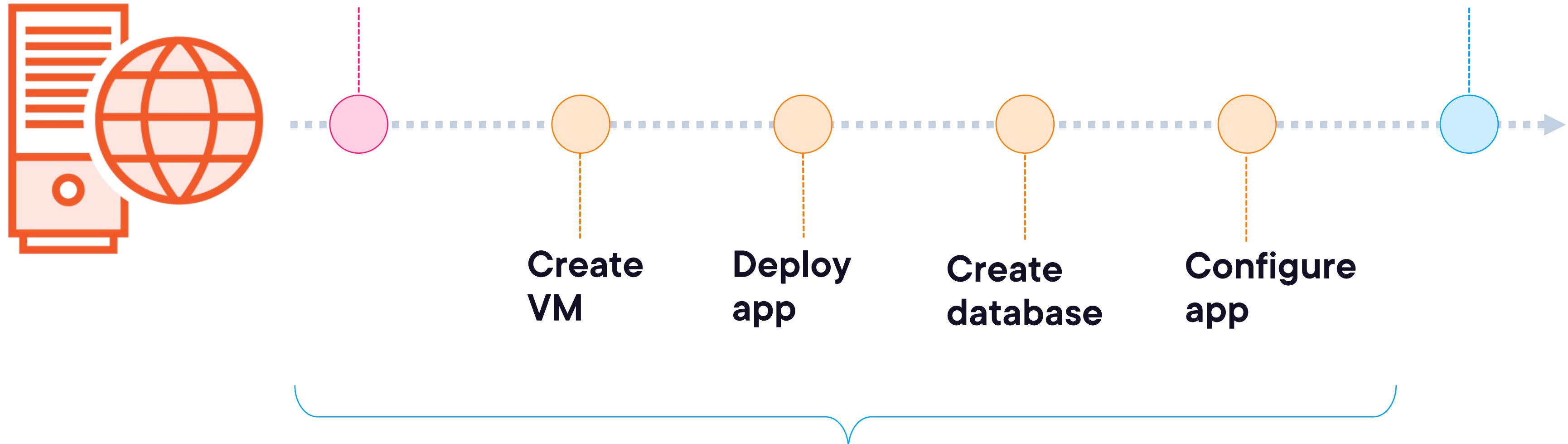
- Moderate systems engineering
- Platform standardization
- Capacity & health benefits



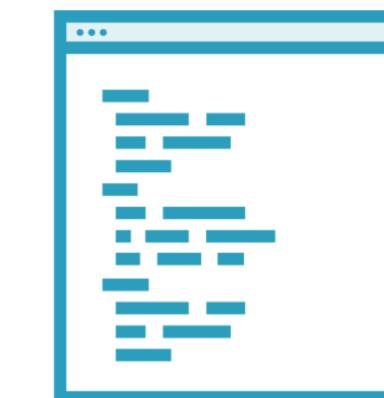
#2?

New env
request

Automate?



- Standard container deployment
- Self-service
- Reusable approach



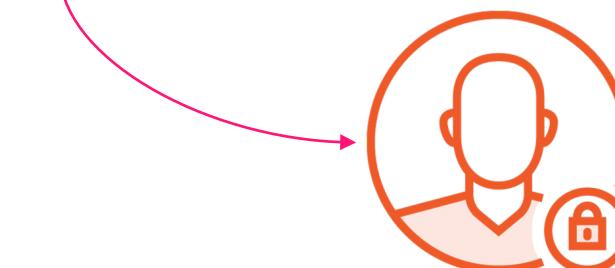
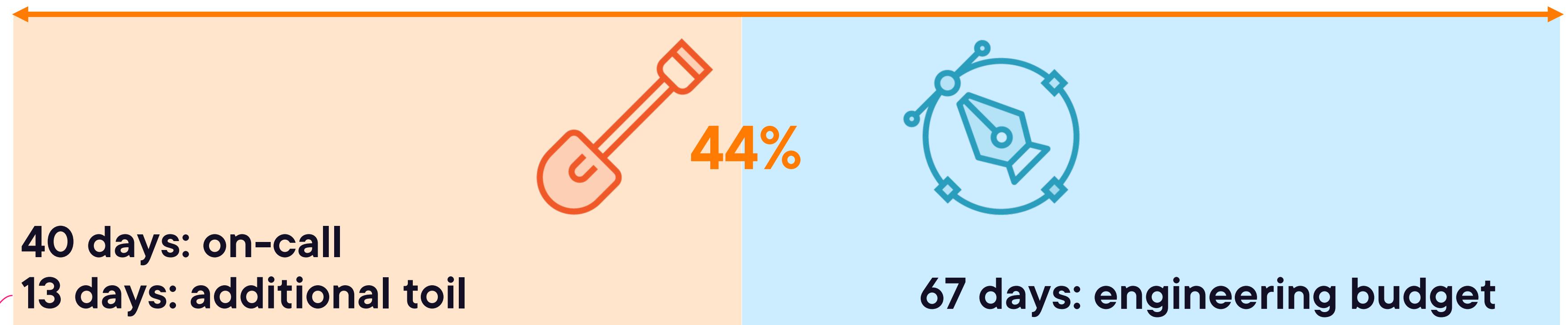
Top Toil Projects

		<i>Toil Days /month</i>	<i>Project Estimate</i>
#1		Patching	4 40
#2		Provisioning	3 5
#3		On-boarding	5 65

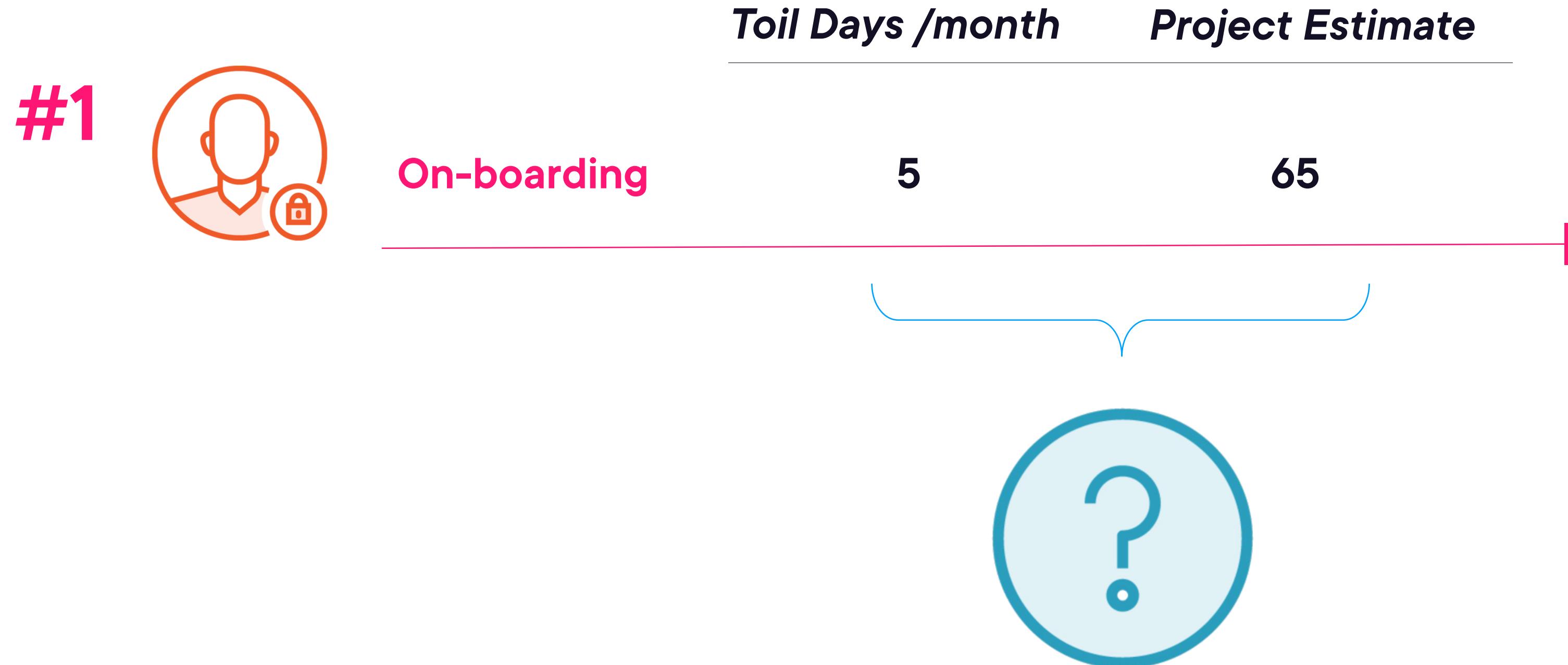


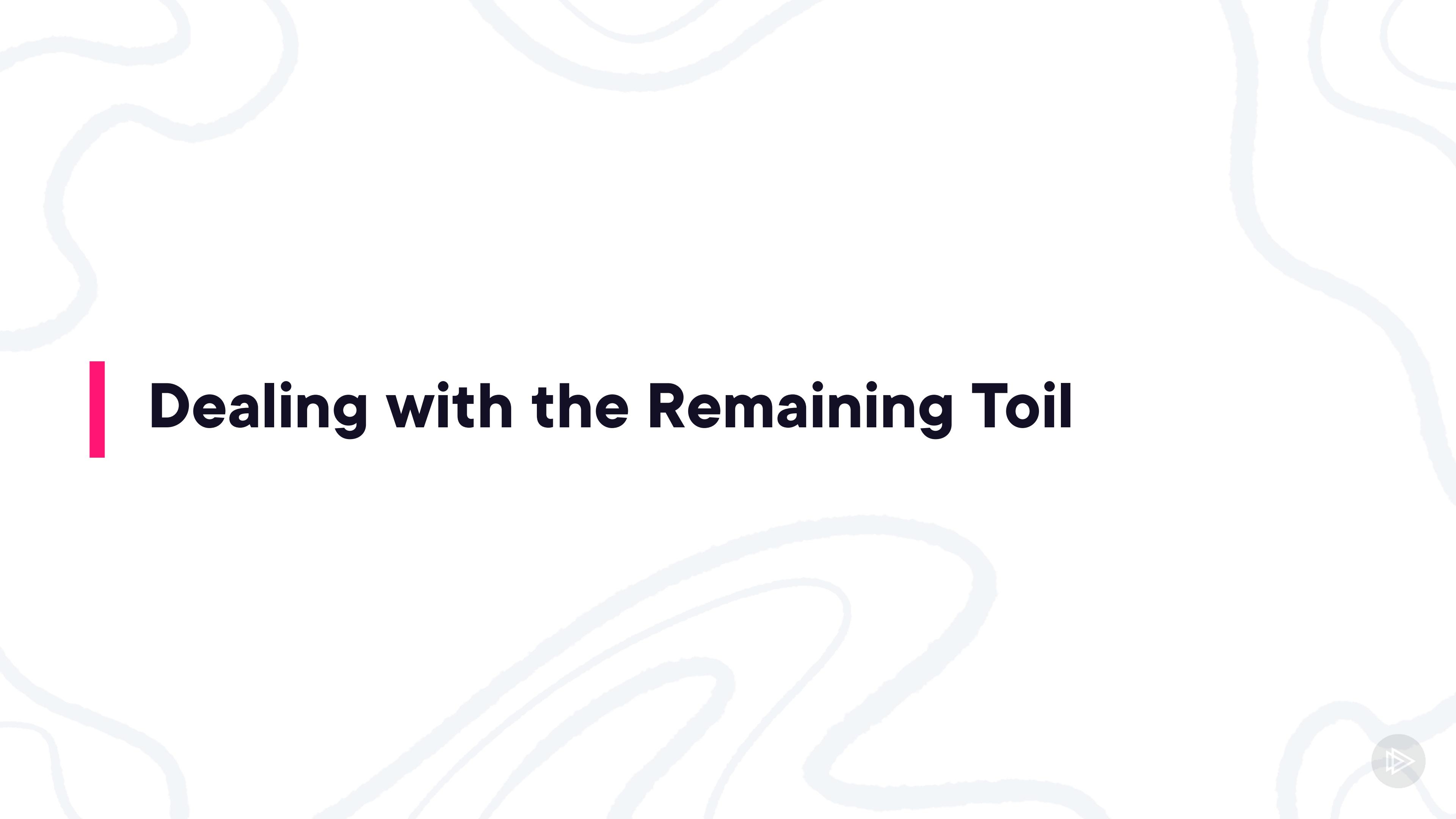


20 working days x6 SREs = 120 days



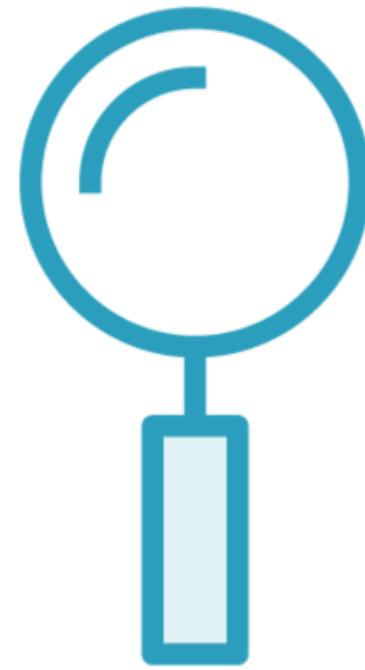
Top Toil Projects



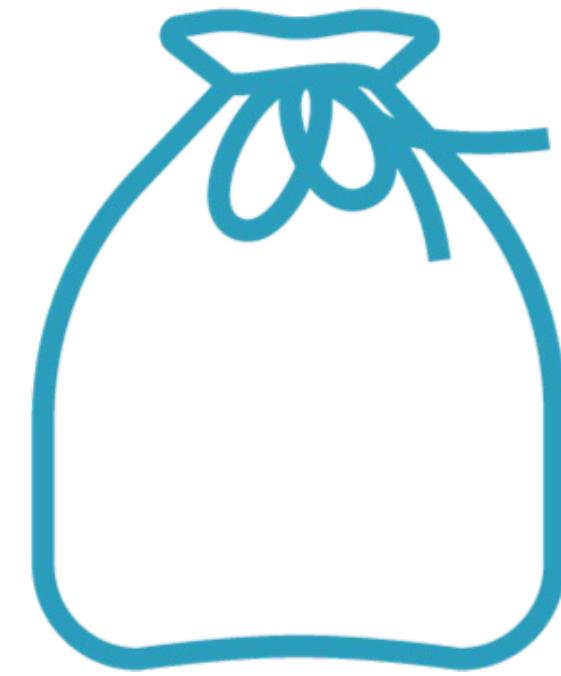


Dealing with the Remaining Toil

Impact Reduction Techniques



Identify & Measure



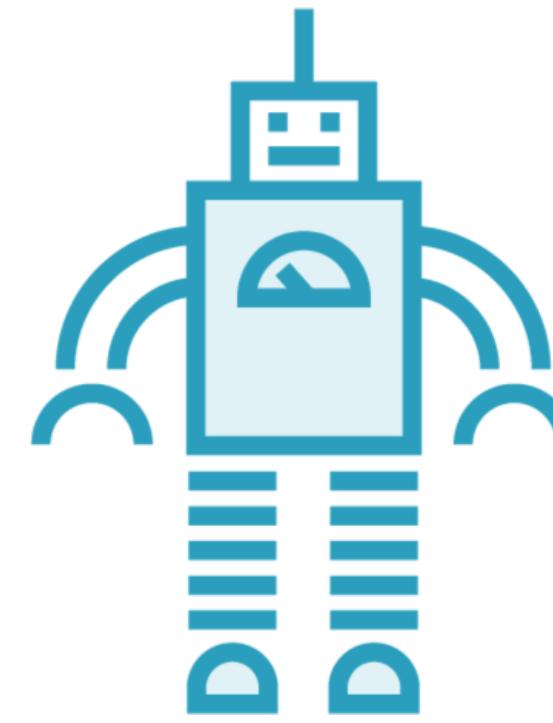
Batch Up



Ignore



Impact Reduction Techniques



Service Facade



Self-Service



Uniformity



Summary



Understanding toil

- Repetitive, low-value work
- Reduces high-value work time

Limiting toil

- Upper limit guarantee (50%)
- Constantly work to reduce

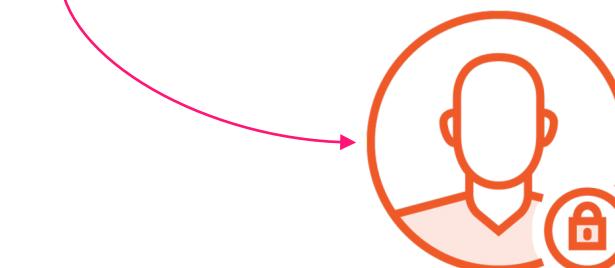
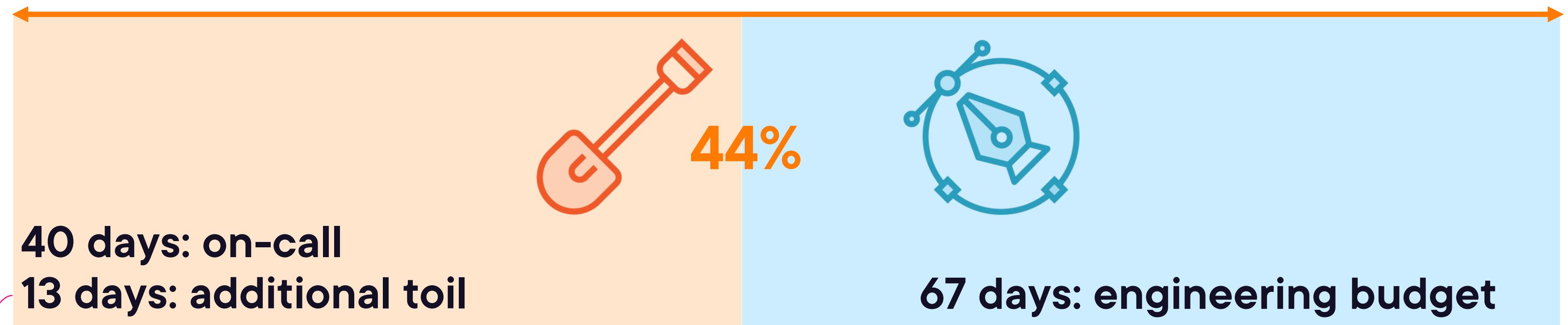
Eliminating toil

- Identify & measure
- Prioritized toil-reduction projects
- Adopt toil-reduction techniques





20 working days x6 SREs = 120 days





20 working days x6 SREs = 120 days



Up Next:

Service Levels, Monitoring and Alerting

