{mthree}

Methods to Eliminate Toil

Objectives

In this module, we will look at ways we can eliminate toil from our system.

Learning Objectives

- Describe when to eliminate toil Describe
- what automation is Identify when to automate Identify where to automate Know when to keep it manual Effective manual toil

{mthree}

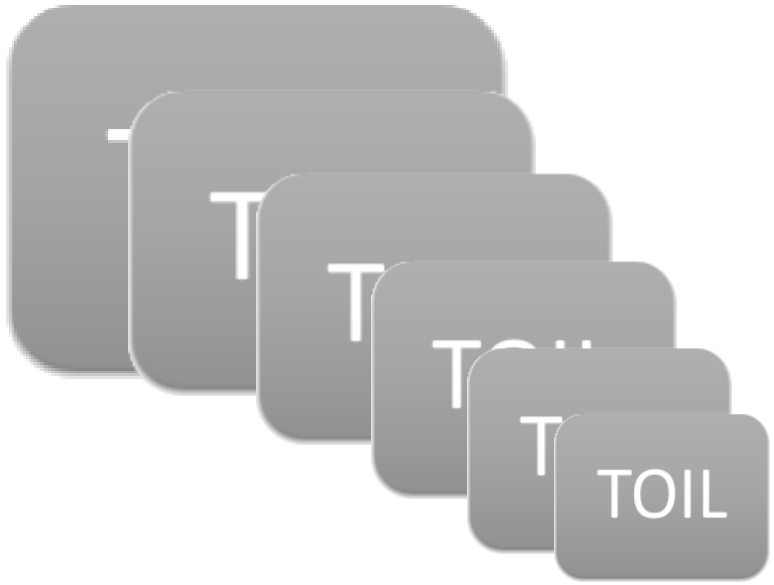
Eliminating toil

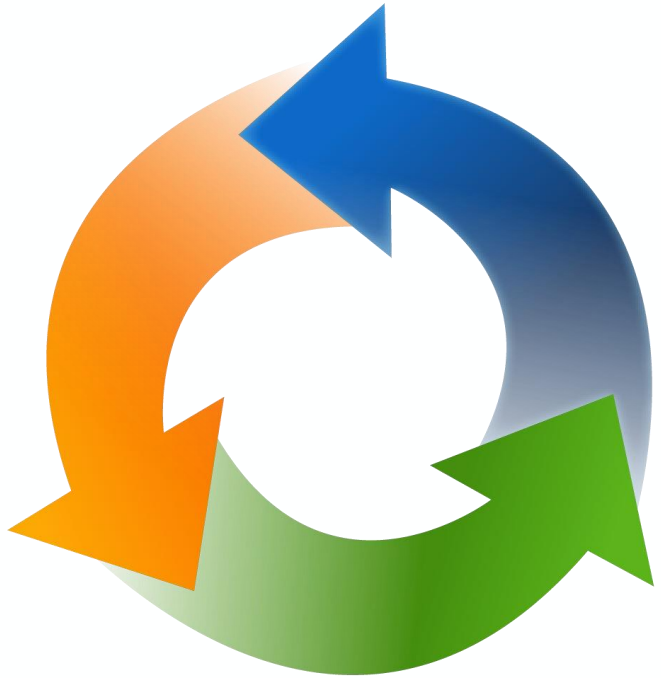
- DevOps
- Automate everything
- SRE
- Does the cost and risk of automating outweigh the manual toil?
- Is the toil well documented?
- Is there team stagnation or low moral Operational tasks > project work
- To keep on top of toil
- Eliminate a piece of toil each week
- An iterative process

- Eliminate through
- Automation (includes the developer, not just DevOps or support)
- – Shift left - problem spotted by PS automated/coded by Devs
- Including into the project backlog
- Improved alerts through categorisation and reducing human requirement
- Self service tools - get the user to do it

{mthree}

Reducing Toil





{mthree}

Elimination Strategies

- Auto-redemption
- Human should be the last point of contact for any alert Where possible, alerts should have an automated response
- Alert categorization and classification Alerts handled by its priority
- Service alerts over hardware degradation
- Self-service tools
- Off load your tasks to the users
- Source: Singh (2020)

{mthree}

What is Automation?

Perform Repetitive Tasks

Use logic to minimize human interaction

{mthree}

Value of Automation

- Consistency
- Enabling scaling
- Examples – Creating user accounts, DNS entries
- Platform
- Extendable to more systems Centralized bug fixing
- Faster repairs
- Regular automation runs can reduce MTTR (Mean Time to Resolution)
- Faster actions
- Application failover, for example
- Time saving
- May not always be obvious
- Enabling other people to run the task rather than just one person = time saving

{mthree}

When to Automate

"Automate yourself out of a job every 18 months"

"[Scaling requires] systems that can self- diagnose and heal and only tell a human it's broken if a human needs to intervene"

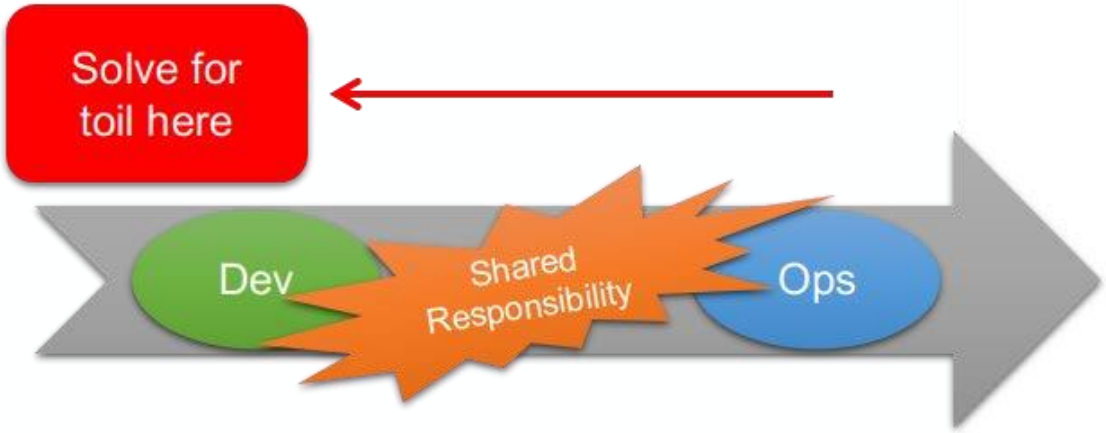
Dave O'Connor

Google

Source: O'Brien (2018)

{mthree}

Shift Left



Dev	DevOps	SRE	Ops
Application coding Automation within App	Pipelines and tooling Supporting Dev and Ops	External script fixes Should be part of app code What should be automated	Script fixes External to applications Documentation

{mthree}

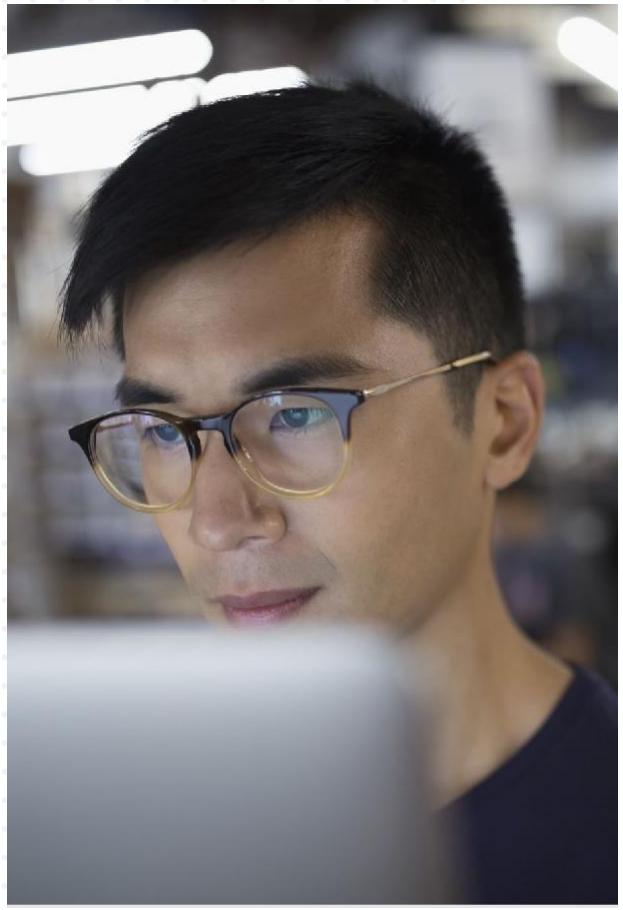
Automation Use Cases

- User account creation
- Cluster turnup and turndown for services
- Software or hardware installation preparation and decommissioning

Rollouts of new software versions

- Runtime configuration changes
- Adding resource to cope with load (autoscaling the service)

Ad infinitum



{mthree}

Activity: When to Automate: Scenario 1

- A report of the number of transactions is required each month
- The report requires a SQL command to be executed in the database It takes 1 minute to type in the command
- The report takes 5 minutes to complete
- The file generated is then emailed to the client
- An engineer identified that it will take 10 minutes to write a script to do this It will take another engineer 10 minutes total to add to Autosys
- However, the report goes to a different person each month
- Do we automate?
- What are your reasons?

{mthree}

Summary: When to Automate

- When you have clearly defined logic that is repeatable
- When the cost of creating the automation is less than performing Is the manual task simple and no stress?
- How often is it being done?
- When there is a benefit to the business
- You may have created something other teams will benefit from This is a business cost benefit in time

{mthree}

Keeping it Manual

- Manual toil tasks should be Simple
- Consistent
- Well documented Proven
- Performing the actions should Cost less than the
- technical debt
- error budget
- Not induce stress on any team member

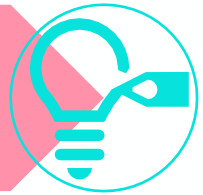
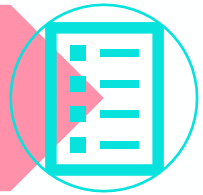
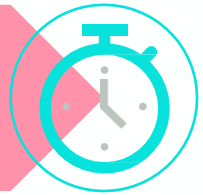
{mthree}

Well-Documented

- Any manual task should be well-defined
- Exact steps to take
- Decisions that need to be made and the resolution Knowledge-based systems required
- Sharing across silos and teams
- Typical tools ServiceNow Jira
- Git
- Tools should allow for direct links to documents

{mthree}

Effective Manual Toil

- Good, clear documentation 
- Concise documentation 
- Anyone can perform the actions 
- Minimal hands-on time 

{mthree}

Activity

- Looking at your site (pipeline, environments and day-to-day tasks) What could be automated that isn't already?
- What would be better remaining as a manual task? What needs to be documented?

{mthree}

Summary
Q & A

{mthree}

References

- Singh, A. K. (2020, January 21). How to Reduce Operations Toil for Site Reliability Engineers. Retrieved from https://medium.com/adobetech/how-to-reduce-operations-toil-for-site-reliability-engineers-acb6761277d5

- O'Brien, C. (2018, July 5). You should automate yourself out of a job every 18 months. Retrieved from https://www.irishtimes.com/business/technology/you-should-automate-yourself-out-of-a-job-every-18-months- 1.3552629

- Murphy, N., Looney, J., and Karirek, M. (2017). Chapter 7 - The Evolution of Automation at Google. Retrieved from https://sre.google/sre-book/automation-at-google/

{mthree}