Laura Nolan, Slack
Twitter: @lauralifts

About This Talk

This talk is about assessing our options when we're running a software system that has issues (cost, stability, excessive toil).

Options

- Turndown
- Migration
- In-Place Modernisation
- Do Nothing and Live With It

TL;DR: IT DEPENDS

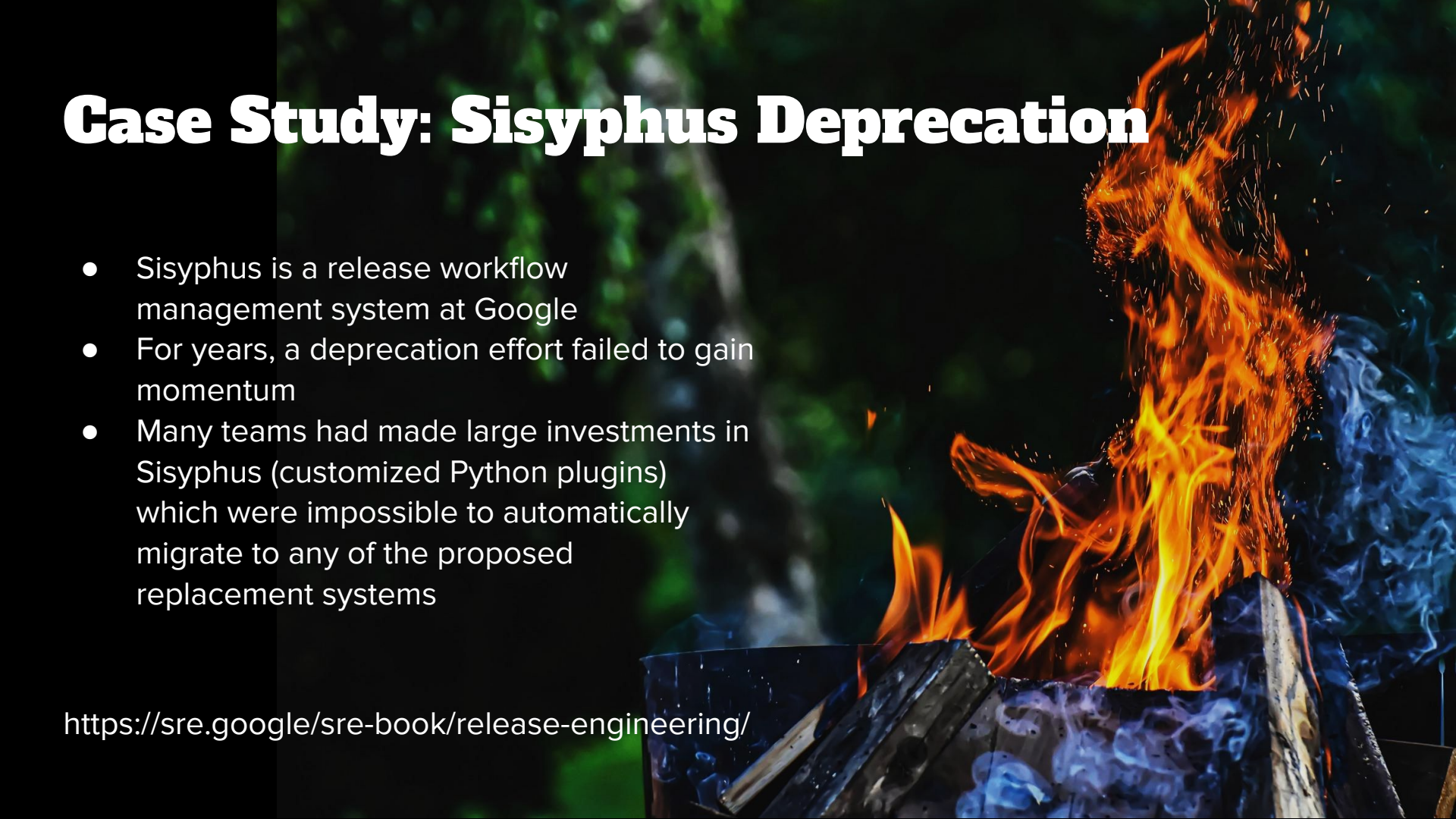
This is a high-consequence type of decision, that can shape the work of teams for years and have effects all throughout an organisation.

Case Study: Photon

- A successful migration from a single-homed data pipeline to a multi-homed fault tolerant architecture
- Desired goals could not be achieved with existing architecture
- Users were not impacted during migration
- Team was staffed adequately
- TOIL went down, reliability increased

https://research.google/pubs/pub41318/

Case Study: Sisyphus Deprecation



- Sisyphus is a release workflow management system at Google
- For years, a deprecation effort failed to gain momentum
- Many teams had made large investments in Sisyphus (customized Python plugins) which were impossible to automatically migrate to any of the proposed replacement systems

https://sre.google/sre-book/release-engineering/

Case Study: HAProxy to Envoy Proxy

- Successful replacement of one OSS proxy with another, with goal of increasing operability and reliability and accessing Envoy-specific features
- Team had sufficient staffing and time
- Impact on users and other teams was minimal
- Gradual migration mitigated risk
- Goals of migration were met

https://slack.engineering/migrating-millions-of-concurrent-websockets-to-envoy/

Case Study: gRPC Streaming in Consul

- The Consul service discovery system uses a protocol called MsgPack
- It recently added support for gRPC streaming for certain blocking read operations which can consume high bandwidth
- With gRPC streaming, a ‘diff’ can be transmitted, rather than the entire dataset
- This is an effective modernisation-in-place to increase scalability

https://www.consul.io/docs/release-notes/1-9-0

How To Make Difficult Choices

1. Ask a question
2. Gather relevant facts and perspectives
3. Articulate your principles
4. Consider how problems may be mitigated
5. Make the decision

The Question

This is probably the easiest step. Your question is something like ‘should we replace Foobar System?’

Relevant Facts and Perspectives

May include:

- Problems with current system
- Feasible alternatives and gap analyses
- Team capacity
- Opportunity costs
- Organisational capacity and ongoing major migrations
- Impact on users
- Risk of migration

Articulate Your Principles

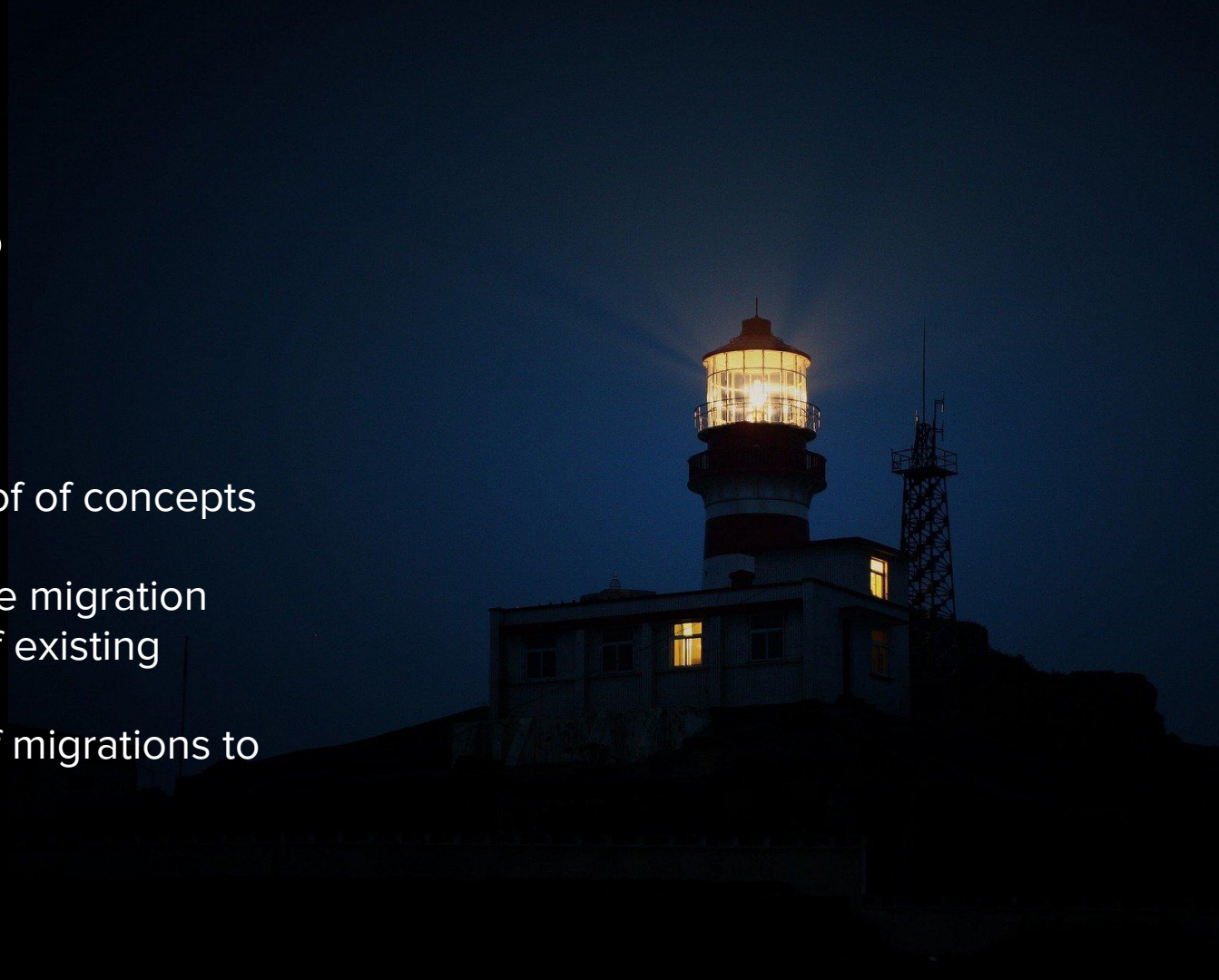
Examples:

- Team workloads should be sustainable
- Systems should not be toilsome
- Users should not be impacted
- An organisation should limit technical churn and risk

Consider Mitigations

Examples:

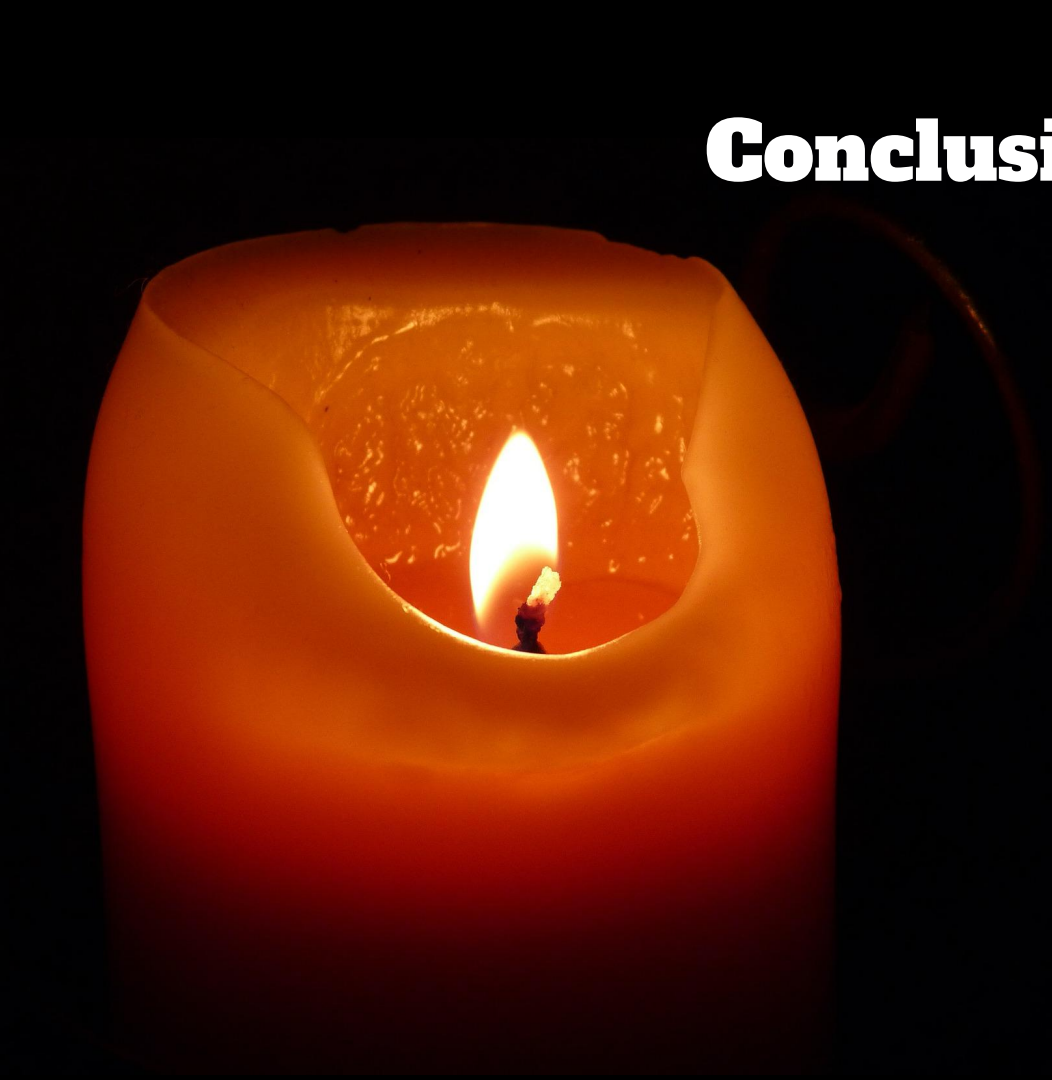
- Prototyping or proof of concepts
- Additional staffing
- Tooling to automate migration processes or toil of existing system
- Careful planning of migrations to reduce risks



Make the Decision

- Clearly document the rationale
- Get people on board
- Revisit if the facts or the principles change

Conclusions



Turndowns and migrations are big and impactful decisions.

Don’t underestimate how long migrations and turndowns take and how costly they are.

Gather facts and perspectives, articulate your values, consider mitigations, and make a decision.