Service Reliability Engineering

Failures are Always an Option

Svilen Ivanov

Sep 16, 2018

HackConf 2018

smule

Let's music together

About

About

- Who am I?

About

- Who am I?
  - curious software engineer

About

- Who am I?
  - curious software engineer

> ...but what I do have are a very particular set of skills. Skills I have acquired over a very long career...
>
> —Liam Neeson, Taken

About

- Who am I?
  - curious software engineer

About

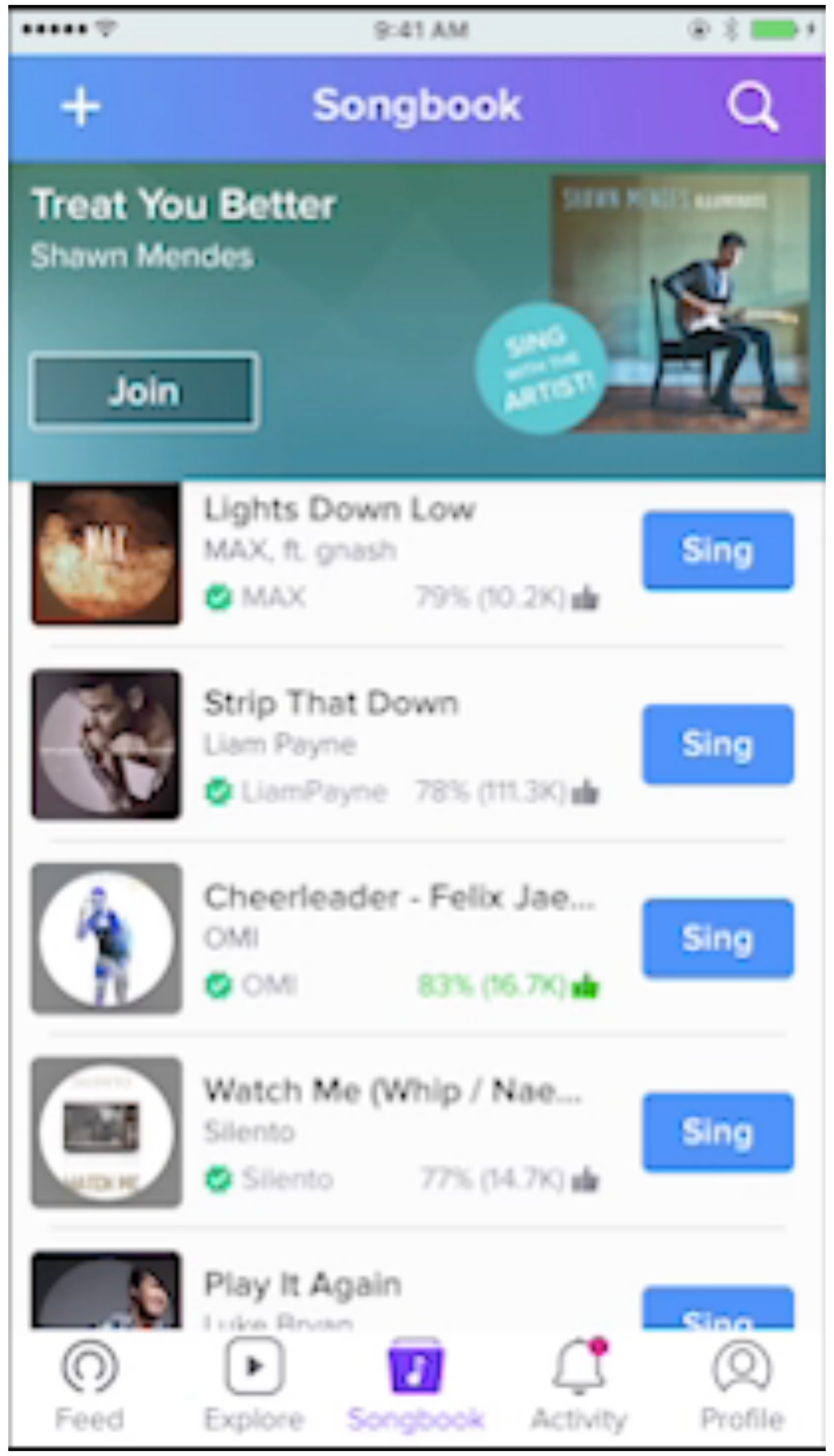
- Who am I?
  - curious software engineer
  - joined Smule 6 years ago

About

- Who am I?
  - curious software engineer
  - joined Smule 6 years ago
- What is Smule?

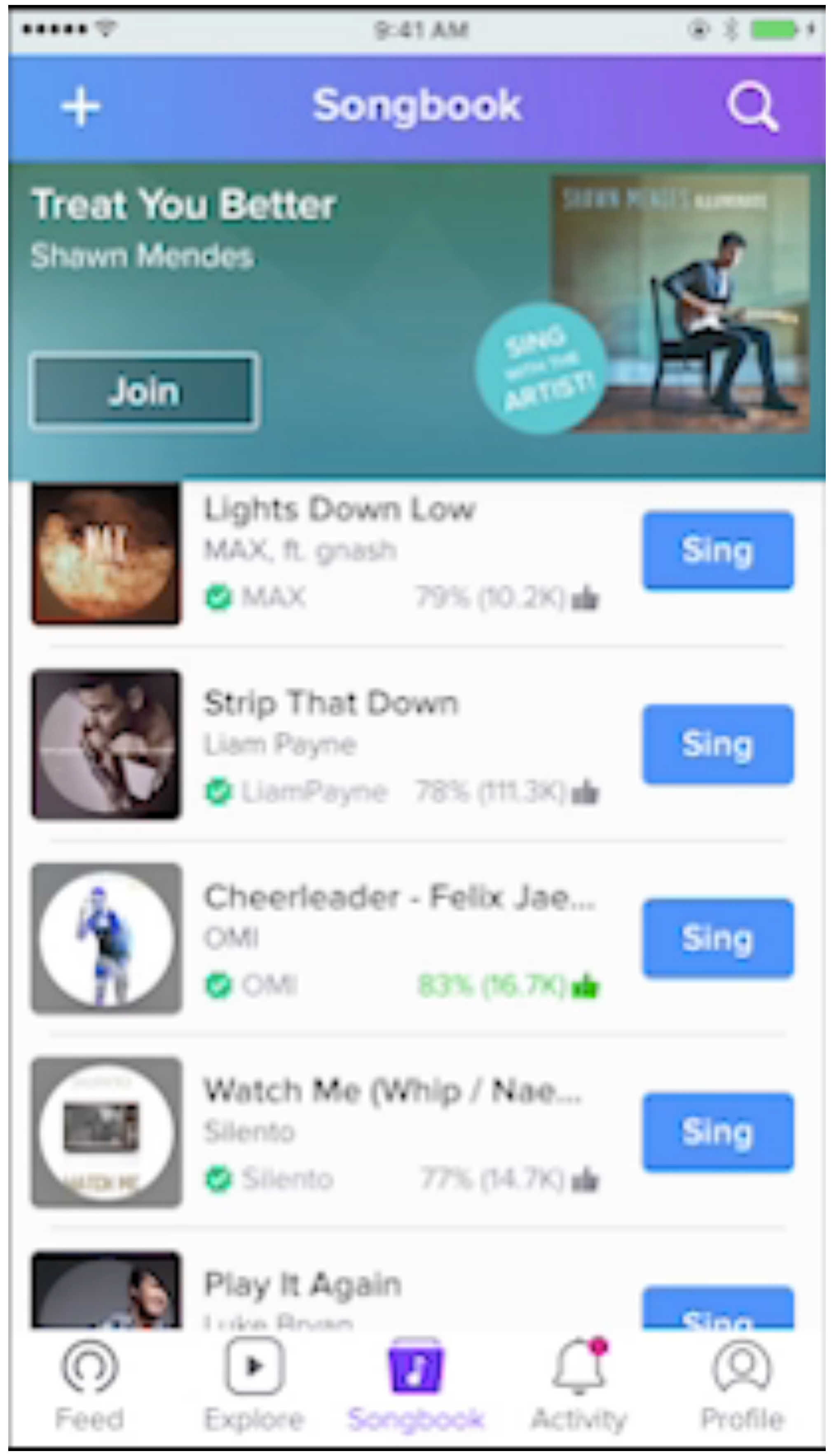
About

- Who am I?
    - curious software engineer
    - joined Smule 6 years ago
- What is Smule?



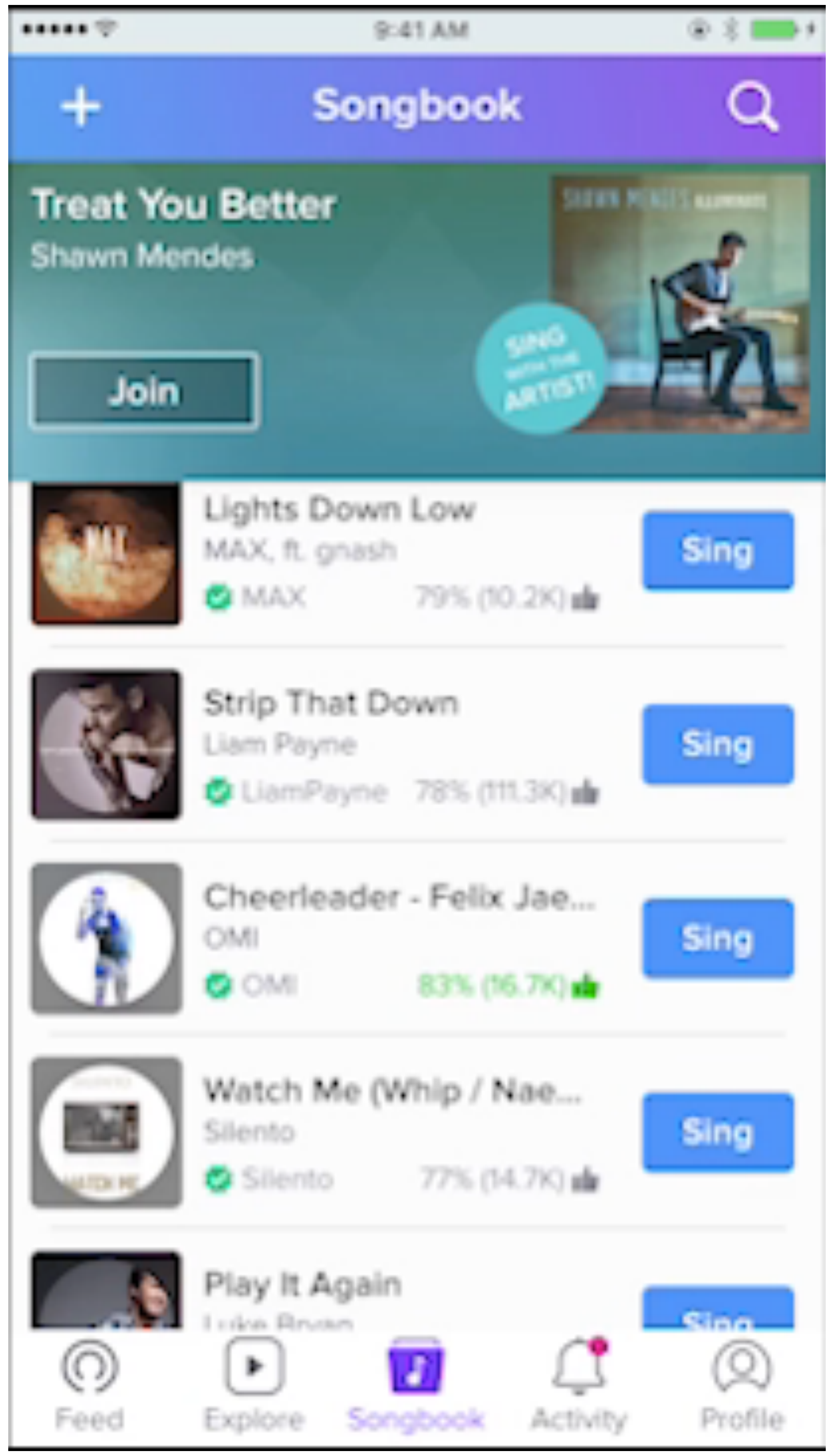
About

- Who am I?
  - curious software engineer
  - joined Smule 6 years ago
- What is Smule?
  - collaborative singing app



About

- Who am I?
  - curious software engineer
  - joined Smule 6 years ago
- What is Smule?
  - collaborative singing app
  - “turn-based” or “realtime”

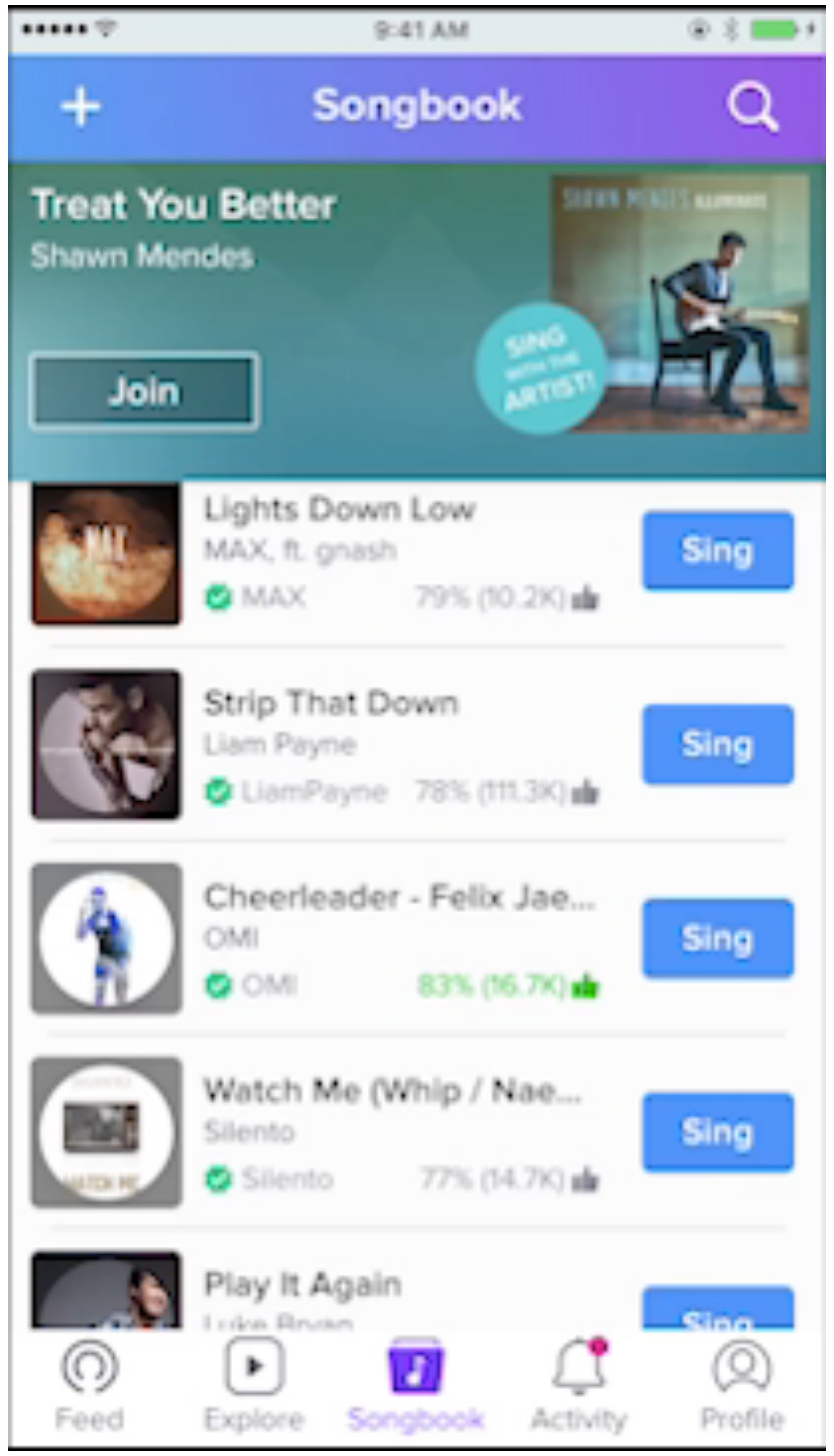


HackConf 2018

smule
Let's music together

About

- Who am I?
  - curious software engineer
  - joined Smule 6 years ago
- What is Smule?
  - collaborative singing app
  - “turn-based” or “realtime”
  - community-driven



9:41 AM

Songbook

Treat You Better
Shawn Mendes

Lights Down Low
MAX, ft. gnash
MAX 79% (10.2K)

Strip That Down
Liam Payne
LiamPayne 78% (111.3K)

Cheerleader - Felix Jae...
OMI
OMI 83% (16.7K)

Watch Me (Whip / Nae...
Silento
Silento 77% (14.7K)

Play It Again
Luke Bryan

Stats

- 50M monthly active users
- 20M songs sang daily
- 20TB data uploaded every day
- 20K requests per second
- 3 geographically distributed datacenters
- 1 mission - connecting the world through music

Production: Expectation

Production: Reality

https://www.youtube.com/watch?v=CRXNCOE7QsA

Production: Reality

https://www.youtube.com/watch?v=CRXNCOE7QsA

What is SRE?

- **Service** - the value you provide to the people
  - a website
  - an app
- **Reliability** - keep the service running
  - one of the most important features of your service
  - it doesn't matter how awesome your app is if no one can use it
- **Engineering** - use your software engineering skills to make the service reliable

Reliability

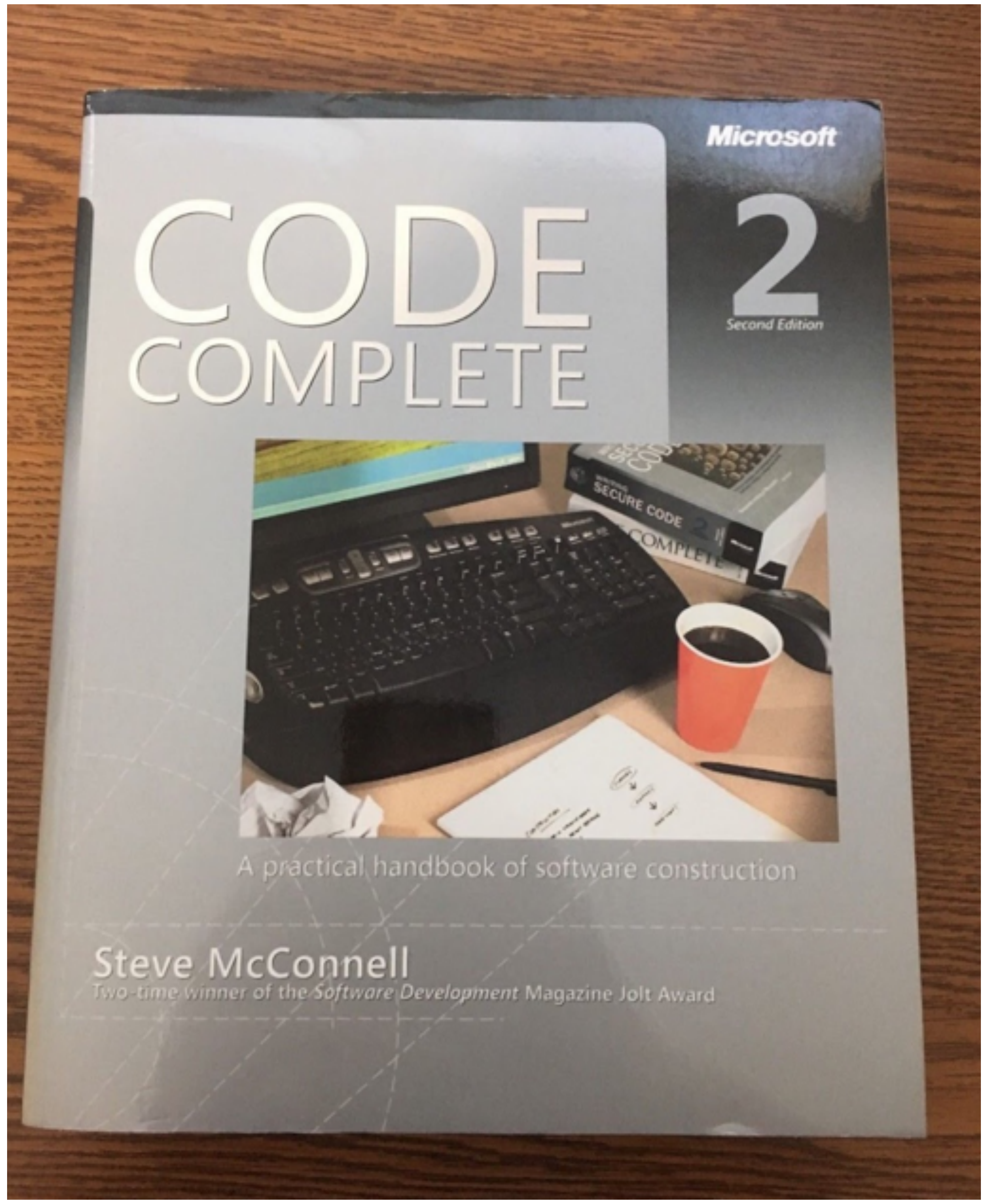
- What decreases the reliability?
  - complexity and dependencies
  - changes
  - project age
- Can you achieve 100% reliability?
  - very expensive
  - contradicts with other requirements (e.g. faster feature development)
  - probably is not needed
- Failures are inevitable

What are Failures

- What can go wrong?
  - everything!
- Computers are inherently unreliable
- Resource (CPU, memory, servers) are finite
- Bugs
  - "about X errors per Y lines of delivered code”
  - Both X and Y are > 0

What are Failures

- What can go wrong?
  - everything!
- Computers are inherently unreliable
- Resource (CPU, memory, servers) are finite
- Bugs
  - "about X errors per Y lines of delivered code”
  - Both X and Y are > 0


Microsoft
CODE COMPLETE
2
Second Edition
A practical handbook of software construction
Steve McConnell
Two-time winner of the Software Development Magazine Jolt Award

Monitoring

- Monitor everything
  - hardware (temperature, S.M.A.R.T, power consumption)
  - OS stats (cpu, memory, network, context switches, etc.)
  - runtime environment stats (JVM, haproxy, etc.)
  - application stats (API performance, request rates, total hits)
- Visualize everything
  - charts
  - maps (geo, heat)

Monitoring (cont.)

- Look for periodicity (seasonality)
  - don’t underestimate holidays in different cultures
  - use external service to validate that your service is accessible from around the world
- Listen to your customers
  - help desk requests
  - socials
  - app store reviews

Monitoring at Smule

- SMG - Smule Grapher
  - Custom stats collection and graphing
- Built with Scala/Play2, uses rrdtool for graphing
- Configuration-based
  - plays nicely with Chef (or any other configuration automation tool)
- Provides aggregated views
- Extensible via plugins
- Robust, scalable, battle-tested
- Open source (Apache 2 license)
  - https://github.com/asen/smg

Alerting

- Setup thresholds
  - conservatively re-evaluate the threshold value
- Categorize alerts
  - informative/warnings
  - critical
- Alerts must be actionable
  - avoid alert fatigue
- Balance the on-call schedule

Post-mortems

- Post-mortem is document describing an incident
- Written shortly the storm is over
- Must be “blameless”

Post-mortems (cont.)

- Document sections:
  - Owner and collaborators
  - Executive Incident Summary
  - Timeline
  - Root Cause
  - Impact
  - What Worked
  - What Went Wrong
  - Action Items

Error Budgets

- Service Level Indicator - uptime, error rate, performance
- Service Level Objective - [any SLI] > 99.99%
- Error budget: (1 - SLO) = 0.01%
  - available for “spending”
- Change is #1 cause of outage
  - Launches are big source of changes
- Spend the budget on launches
  - over the budget: pause the feature development to improve the reliability
  - below the budget: launch the feature into production

Resources

- SMG
  - https://github.com/asen/smg
- Site Reliability Engineering Book
  - https://landing.google.com/sre/book/index.html
- Site Reliability Engineering at Google talk by Christof Leng
  - https://youtu.be/Cxb7a8lTv8A

Thank you!
Questions?