



DevOpsGroup

Cloud and DevOps. **Fast, secure and simple.**



Rael Winters

Scale-up Division
DevOpsGroup

We make Cloud and DevOps **fast, secure and simple.**



Cloud and DevOps
Engineering



Training



Consulting



Cloud-Native
Managed Services

> **61%**

Investment for UK scaleup digital tech firms grew 61% between 2017 and 2018.

> **35%**

35% of Europe and Israel's 169 unicorn tech companies have been created in the UK.

> **2.5x**

Scaleup tech investment was 2.5x higher than expected based on the relative size of the UK economy.

> **6x**

Investment in AI grew almost sixfold from 2014 to 2018.

> **£6.3bn**

Total venture capital investment in UK tech in 2018 topped £6bn, more than any other European country.

> **56%**

The recent growth rate for London tech scaleups at 56% makes the cluster first in the world for scaleup growth.

> **4th**

With £5bn of scaleup investment, the UK ranks fourth in the world, after the US, China and India.

> **80%**

Ambitious scaleups are driving UK tech advantage - tech scaleup deals delivered the majority of all tech investments in the UK in 2018 (£5bn of over £6bn).



Most scale-ups aren't killed;
they self-destruct.

Entrepreneurial growth post start-up has numerous challenges, which can be an order of magnitude more difficult than simply starting a venture.

Daniel Isenberg, Harvard Business Review

3%

Only 3% of start-ups survive beyond 10 years

92%

92% of start-ups fail within 3 years; predominantly (74%) due to self-destruction, rather than competition ¹

0.4%

Only 0.4% of Scale-ups reach revenues of £50 to £100 mil without plateauing ²

[1] https://s3.amazonaws.com/startupcompass-public/StartupGenomeReport2_Why_Startups_Fail_v2.pdf

[2] <https://scaleupnation.com/>



Technical Debt

Dealing with decisions and actions from the past, recognising when things have been built badly or in a rush, then repaying the debt.



Modern Operations

Dealing with unplanned work without impacting velocity; and bringing stability without compromising agility.



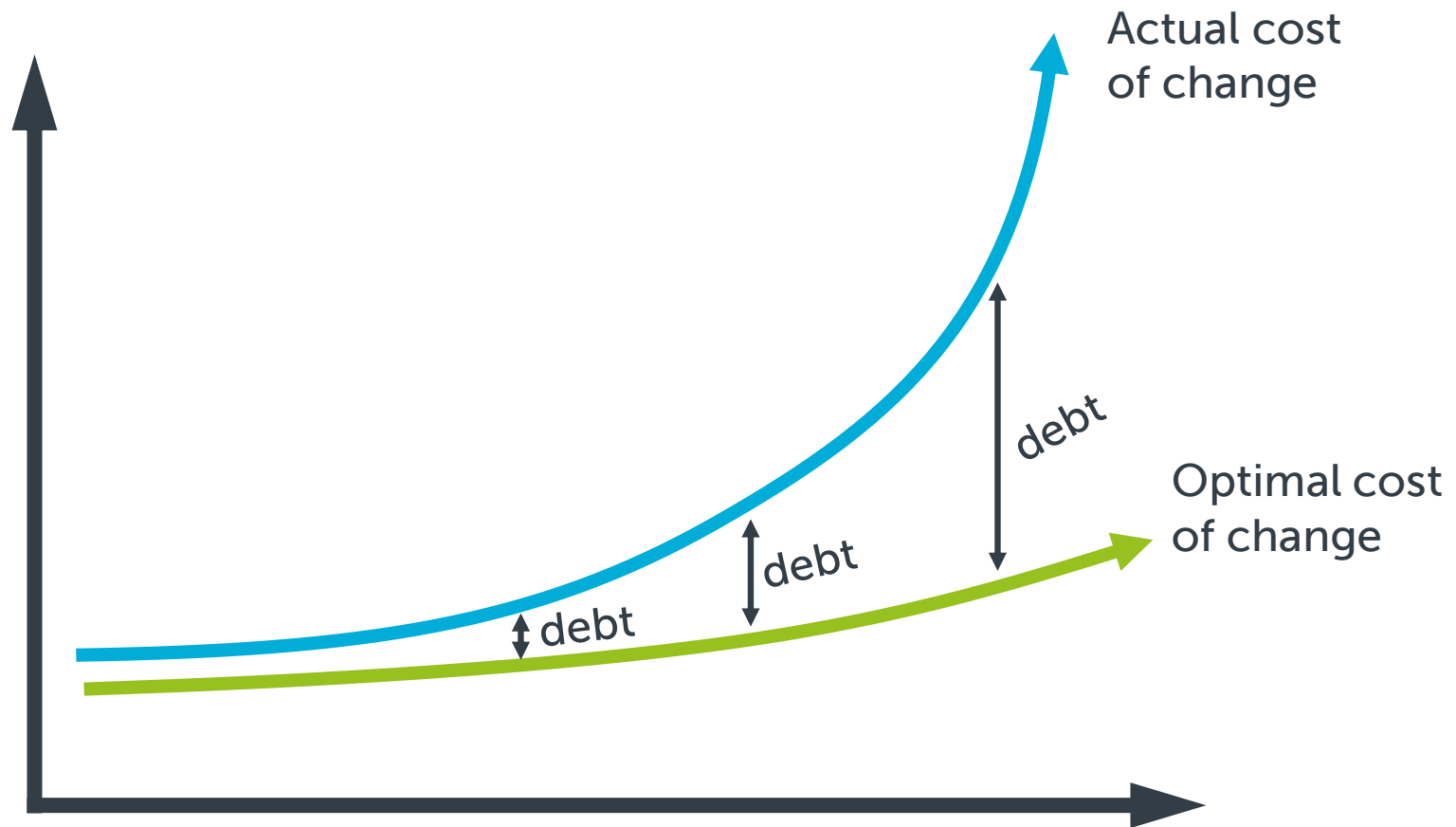
Staying Agile

Enabling faster and safer innovation. Continually realign product-market fit, even when moving at pace with high value at risk.

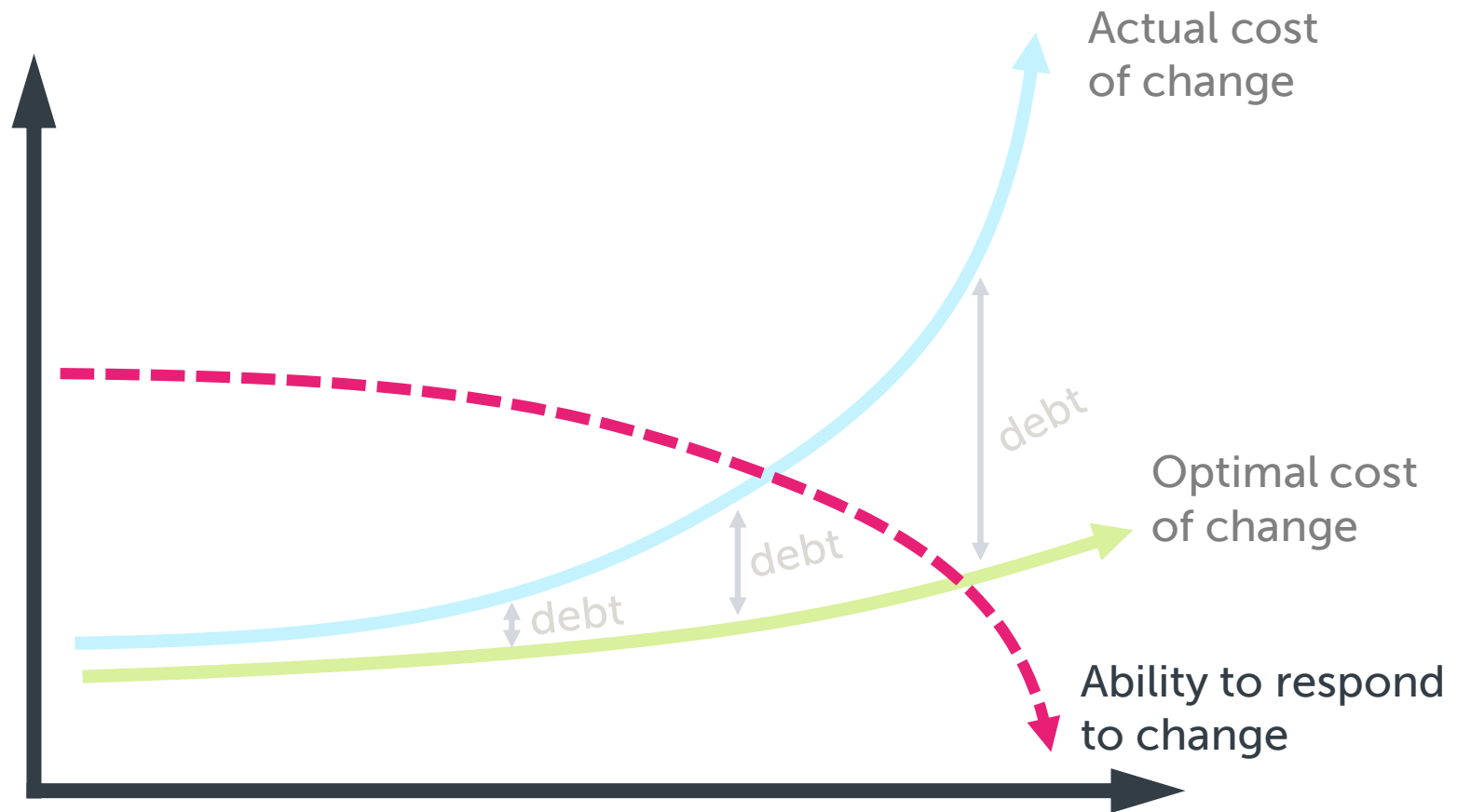




Cost of of
Change



Cost of of Change



Exploration of Technical Debt in Start-ups

Eriks Klotins*
Blekinge Institute of Technology
Karlskrona, Sweden
eriks.klotins@bth.se

Michael Unterkalmsteiner
Blekinge Institute of Technology
Karlskrona, Sweden

Panagiota Chatzipetrou
Blekinge Institute of Technology
Karlskrona, Sweden

Tony Gorschek
Blekinge Institute of Technology
Karlskrona, Sweden

Rafael Prikladnicki
Pontifical Catholic University of Rio
Grande do Sul
Porto Alegre, Brazil

Nirnaya Tripathi
University of Oulu
Oulu, Finland

Leandro Bento Pompermaier
Pontifical Catholic University of Rio
Grande do Sul
Porto Alegre, Brazil

ABSTRACT

Context: Software start-ups are young companies aiming to build and market software-intensive products fast with little resources. Aiming to accelerate time-to-market, start-ups often opt for ad-hoc engineering practices, make shortcuts in product engineering, and accumulate technical debt.

Objective: In this paper we explore to what extent precedents, dimensions and outcomes associated with technical debt are prevalent in start-ups.

Method: We apply a case survey method to identify aspects of technical debt and contextual information characterizing the engineering context in start-ups.

Results: By analyzing responses from 86 start-up cases we found that start-ups accumulate most technical debt in the testing dimension, despite attempts to automate testing. Furthermore, we found that start-up team size and experience is a leading precedent for accumulating technical debt: larger teams face more challenges in keeping the debt under control.

Conclusions: This study highlights the necessity to monitor levels of technical debt and to preemptively introduce practices to keep the debt under control. Asking more people to an already difficult to maintain product could amplify other precedents, such as resource shortages, communication issues and negatively affect decisions pertaining to the use of good engineering practices.

KEYWORDS

Software start-ups, technical debt

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICSE/SEIP '18, May 27–June 3, 2018, Gothenburg, Sweden
© 2018 Association for Computing Machinery
ACM ISBN 978-1-60559-619-9/\$15.00
<https://doi.org/10.1145/3183519.3183539>

ACM Reference Format:

Eriks Klotins, Michael Unterkalmsteiner, Panagiota Chatzipetrou, Tony Gorschek, Rafael Prikladnicki, Nirnaya Tripathi, and Leandro Bento Pompermaier. 2018. Exploration of Technical Debt in Start-ups. In *ICSE/SEIP '18: 40th International Conference on Software Engineering: Software Engineering in Practice Track*, May 27–June 3, 2018, Gothenburg, Sweden. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3183519.3183539>

1 INTRODUCTION

Start-ups are important suppliers of innovation, new software products and services. However, engineering the software in start-ups is a complex endeavor as the start-up context poses unique challenges to software engineers [10]. As a result of these challenges, most start-ups do not survive the first few years of operation and cease to exist before delivering any value [4, 11].

Uncertainty, changing goals, limited human resources, extreme time and resource constraints are reported as characteristic to start-ups [10, 27]. To cope with such forces, start-ups make a trade-off between internal product quality and faster time-to-market, favoring the latter. As a consequence, start-ups accumulate technical debt [5].

Technical debt is a metaphor to describe not quite right engineering solutions in a product that adds friction to its further development and maintenance. The extra effort associated with this friction, i.e. the "interest", needs to be paid every time a sub-optimal solution is touched [21]. Over time, the cumulative interest may exceed the effort needed to remove the debt, i.e. the "principal". The compound effects of sub-optimal solutions can reduce development team efficiency and overall quality. However, there is the belief among start-ups that any amount of technical debt can be written off if a feature or the whole product is not successful in the market [5].

The strategy to accumulate technical debt can backfire if a start-up survives long enough and fails to put its technical debt under control. An unstable and difficult to maintain product adds risk to the company, for example, by limiting the ability to quickly enter into new markets (i.e. to pivot [38]) or to launch new innovative features [19, 39]. That said, we are not advocating for the removal of all technical debt. Rather, we are interested to see an overview

①

Technical debt matters. There is an association between technical debt and a start-up outcome. Having less technical debt gives a start-up more room to pivot or evolve the product.

②

Less experienced teams perceive debt less but accrue more. Use external expert help to identify unrealized debt and improve sub-optimal practices.

③

Team size correlates with more severe debt. To support team growth, more coordination practices are needed, and impact on technical debt needs to be monitored.

④

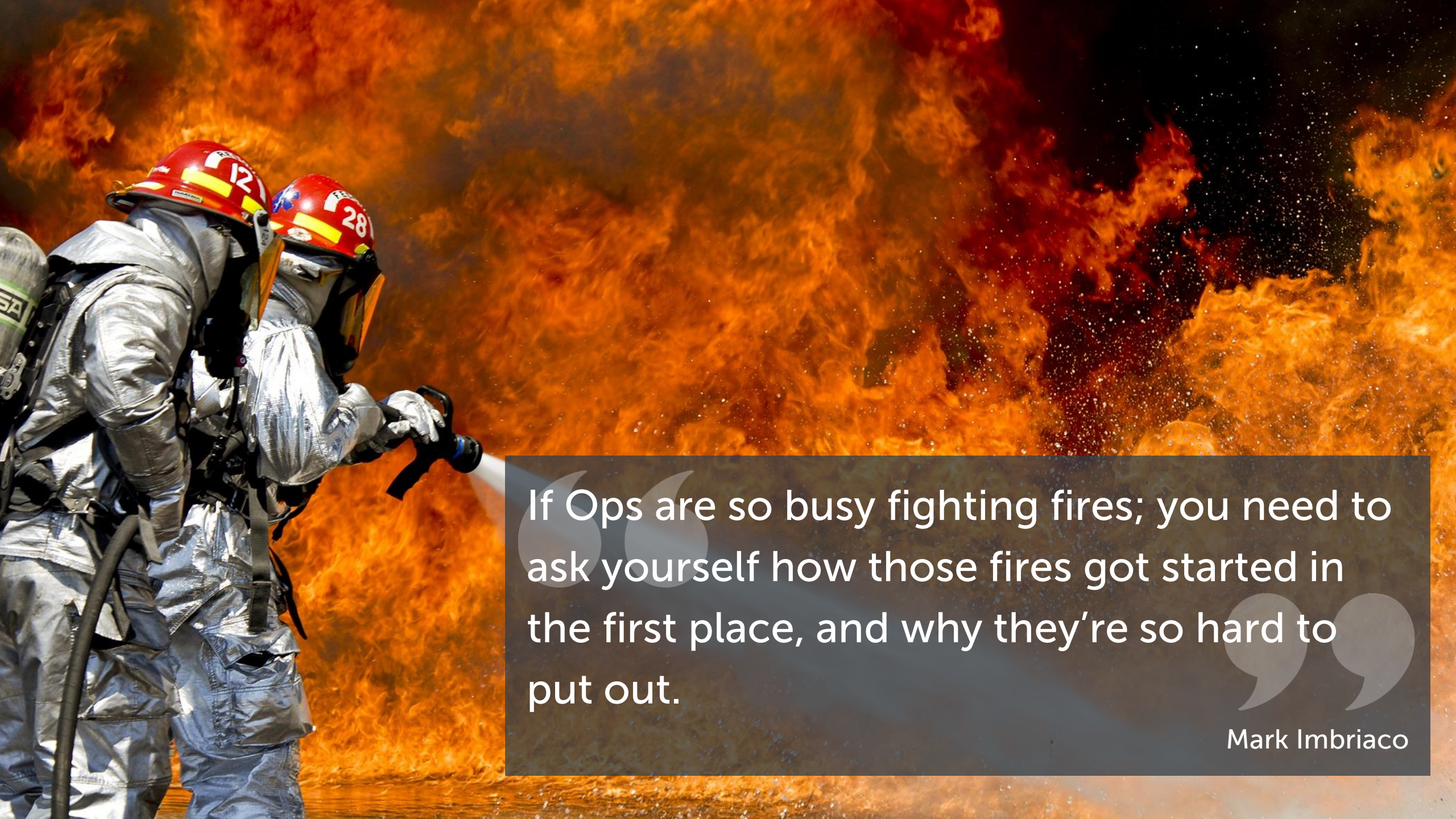
The most significant type of technical debt in start-ups is code smells. Poorly structured and documented code has the strongest association with issues in team productivity and product quality. Detection of code smells can be automated with open-source tools, thus alleviating removal of this type of debt.

Snowflakes

/ˈsnəʊfleɪk/ uniquely configured infrastructure components (virtual machines, containers, networks etc.) that aren't written or configured with code or managed through source control.

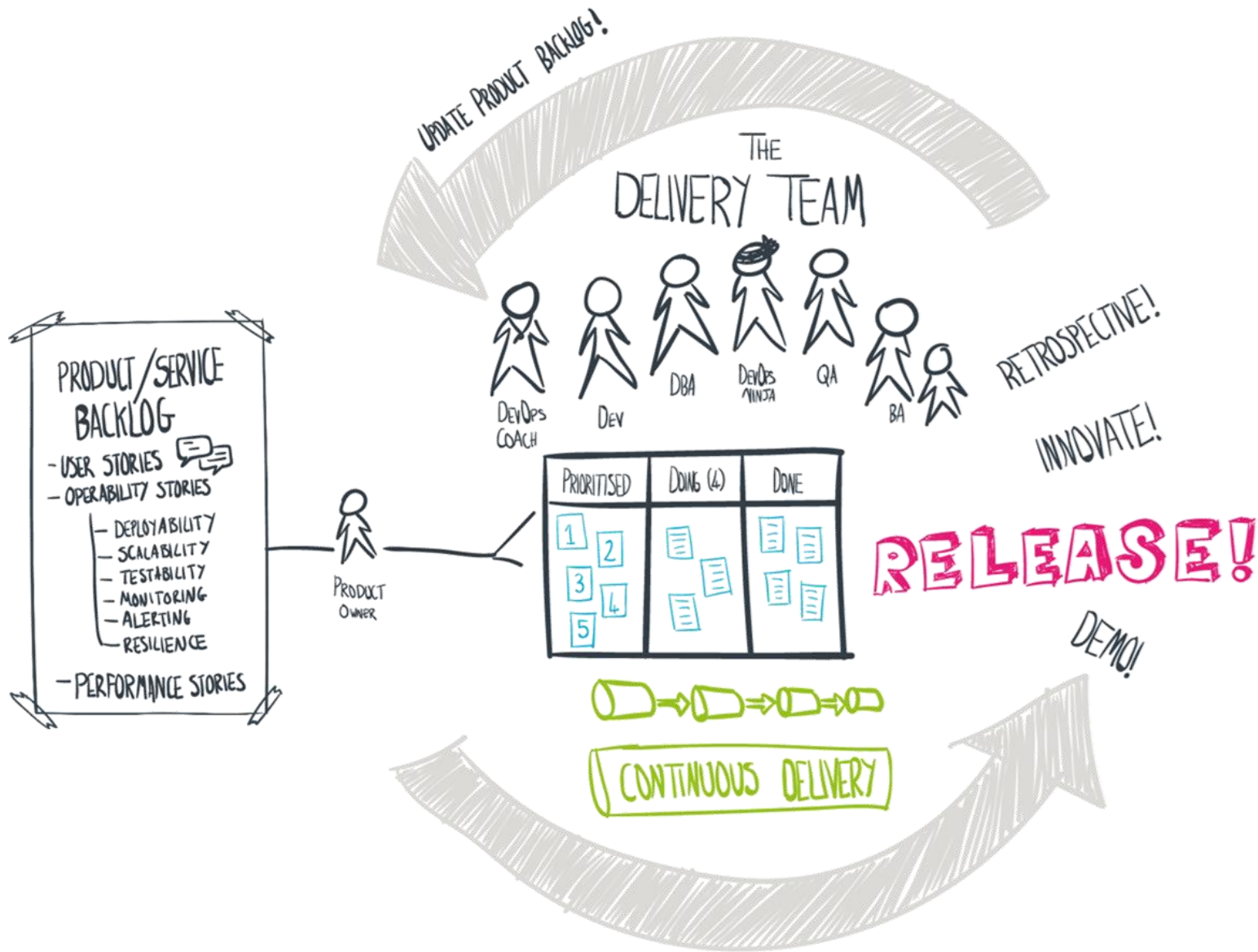
1. Environment creation is usually the first constraint that needs to be resolved as the software delivery process is optimised.¹
2. Teams using infrastructure as code are 1.8x more likely to be 'elite performers'.²





“If Ops are so busy fighting fires; you need to ask yourself how those fires got started in the first place, and why they’re so hard to put out.”

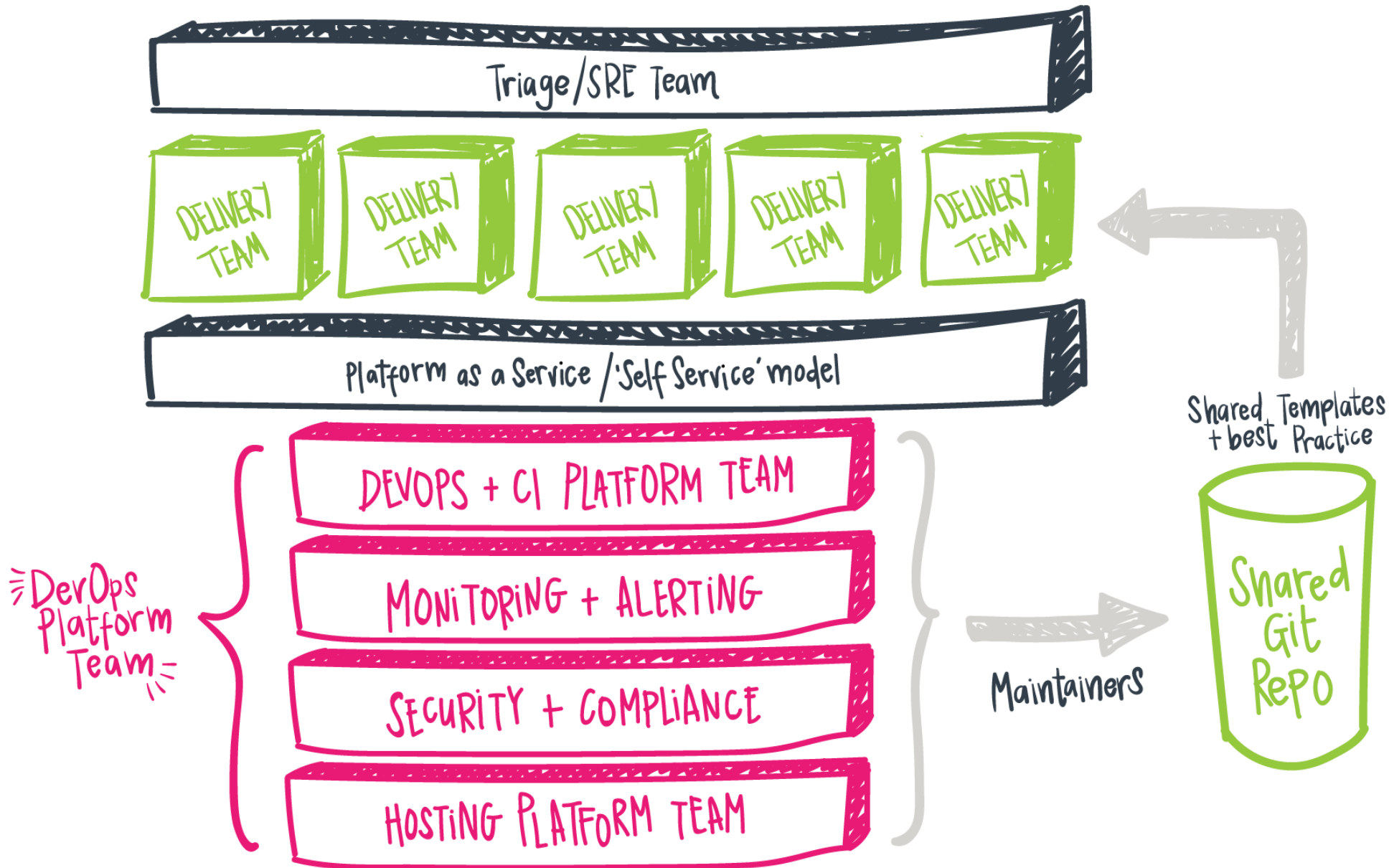
Mark Imbriaco



PRACTICES

- SOURCE CONTROL EVERYTHING!
- TRIAGE UNPLANNED WORK
- PRODUCT BOARD
- STAND-UPS
- AUTOMATED
 - BUILDS
 - TESTING
 - INFRASTRUCTURE
 - DEPLOYMENTS
 - EVERYTHING!
- KANBAN





Technical Debt



- Acknowledge it, understand it
- Track it in the backlog, and track the pain
- Repay the principal

Modern Operations



- Operability features as first-class citizens
- Products, not projects
- Self-service operations

Stay Agile



- Build-Measure-Learn
- Safety comes in small batches
- Two-way doors



An innovative, experimental culture is critical to our success as a fast-growing fintech, and we now have the means to build, deploy and host new products and features to accelerate our growth.

Michael Ashford, CTO and co-founder



DevOpsGroup

Cloud and DevOps. **Fast, secure** and **simple.**



0800 368 7378



team@devopsgroup.com



[@DevOpsGroup](https://twitter.com/DevOpsGroup)



www.devopsgroup.com

