

Continuous Delivery: Benefits Explained

Lianping Chen

lchen@paddypower.com

@lianpingchen

The presentation represents only my own views and doesn't necessarily reflect those of my employer.

PADDYPOWER.



Offers betting/gambling services in regulated markets, through shops, phones, Internet, mobile apps

7 billion turnover

5000 employees

Relies heavily on an increasingly large number of
custom software applications

These applications are developed and maintained by the Technology Department, which employs about 500 people

PADDYPOWER.

betfair



The world's largest public online betting and gaming company with combined revenues of more than £1.1bn

The 44th largest company on the London Stock
Exchange

About four years ago ...

Many releases were a “scary” experience.



Priority 1 incidents

The delivery activities weren't efficient. Just setting up the testing environment could take three weeks.

Each application typically had fewer than six releases a year.

This release model artificially delayed features completed early in the release cycle.

The value these features could generate was
lost.

Early feedback on them wasn't available.

DEVELOPMENT

READY FOR ACCEPTANCE

ACCEPTANCE

DONE!

DEV IN PROGRESS

REVIEW

DELS: [Redacted] PADDYPOWER.
 Mick

DELS: [Redacted] PADDYPOWER.
 LianPing

DELS: [Redacted] PADDYPOWER.
 Mark

DELS: [Redacted] PADDYPOWER.
 Krystof

SMPL 141 PADDYPOWER.
 Build [Redacted]

[Redacted] PADDYPOWER.

SMPL 156 PADDYPOWER.
 D [Redacted]

[Redacted] PADDYPOWER.
 DEV TEAMS CAN CREATE

[Redacted] PADDYPOWER.

[Redacted] PADDYPOWER.
 SMALL CONFIG CHANGES

[Redacted] PADDYPOWER.
 FOR ELECTRIC SHEEP

[Redacted] PADDYPOWER.
 UPLOAD DATABASES

[Redacted] PADDYPOWER.
 DMS PIPELINE FOR
 MASTER & LIVE (PROD)

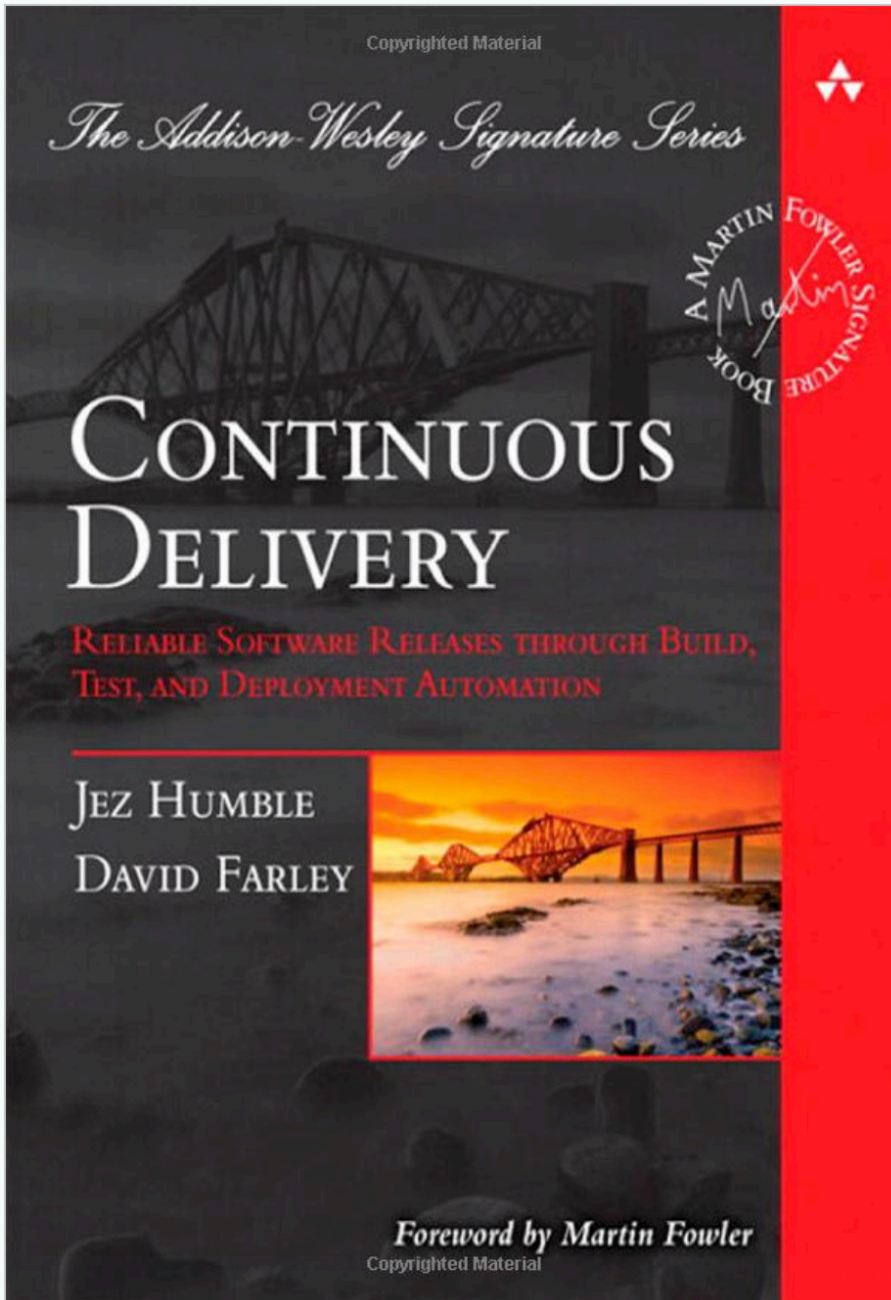
[Redacted] PADDYPOWER.
 COM LOGIN FORCED
 JENKINS

NPM PADDYPOWER.
 [Redacted]

[Redacted] PADDYPOWER.
 Create / Update

DELS: [Redacted] PADDYPOWER.
 ONE-OFF FOR





Does it really work?

To what extent?

What are the actual benefits that have been realized?

“There is more to Continuous Delivery than wiring together Jenkins instances and buying a new automated deployment tool.”

What are those more things?

Reliable Releases

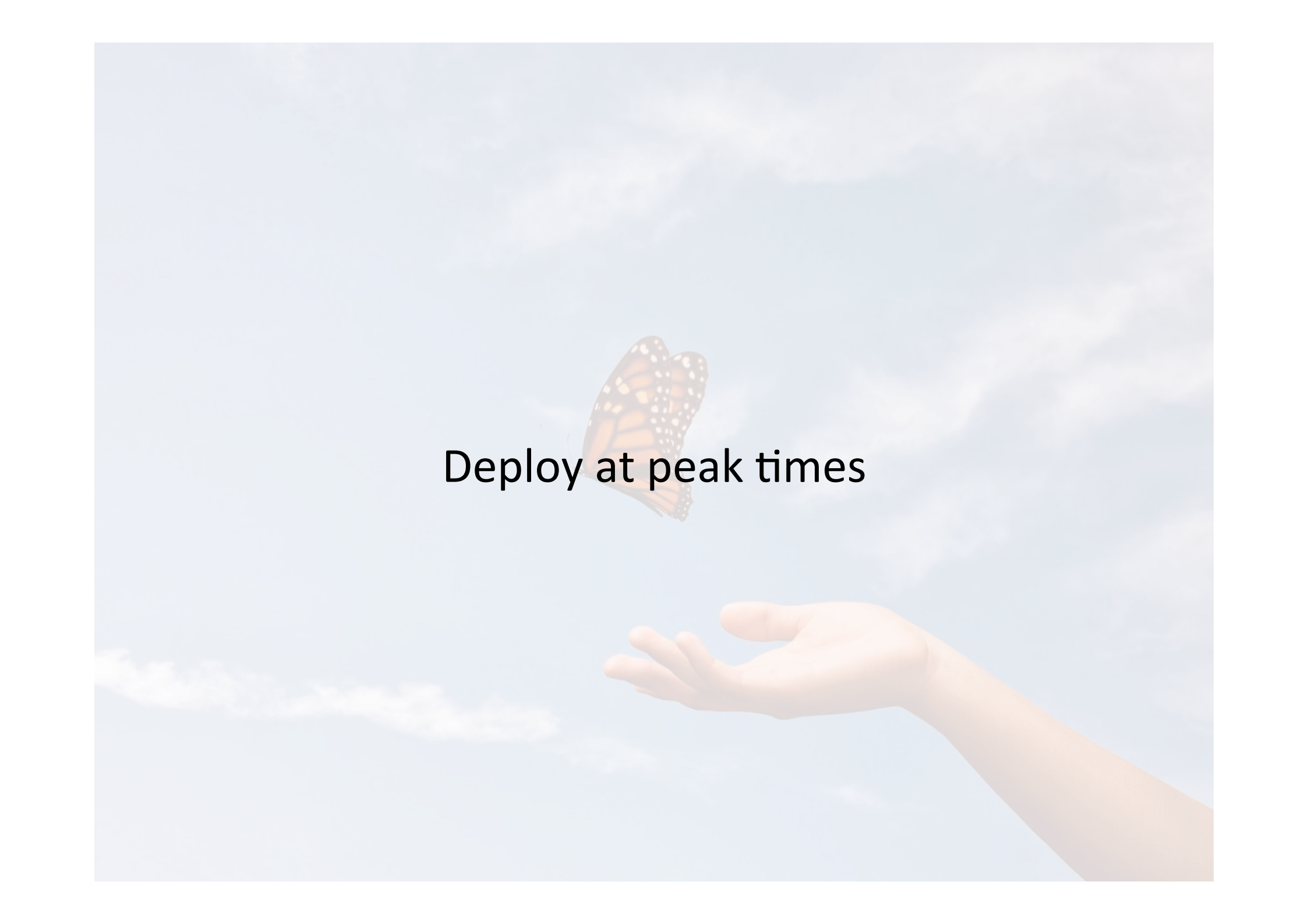


A photograph of a hand holding a monarch butterfly against a cloudy sky. The hand is positioned at the bottom right, with fingers slightly curled. The butterfly is perched on the palm, facing left. The background is a bright blue sky with soft, white clouds. The text is centered in the middle of the image.

The risks associated with a release have significantly decreased

A photograph of a hand releasing a monarch butterfly into a cloudy sky. The hand is positioned at the bottom right, with fingers slightly curled. The butterfly is in the center, with its wings spread, showing orange and black patterns. The sky is light blue with soft, white clouds. The text is centered over the image.


No stress on the release day anymore. That day becomes just another normal day.

A photograph of a hand holding a monarch butterfly against a cloudy sky. The hand is positioned at the bottom right, with fingers slightly curled. The butterfly is in the center, with its wings spread, showing orange and black patterns. The sky is light blue with soft, white clouds. The text "Deploy at peak times" is overlaid in the center of the image.

Deploy at peak times

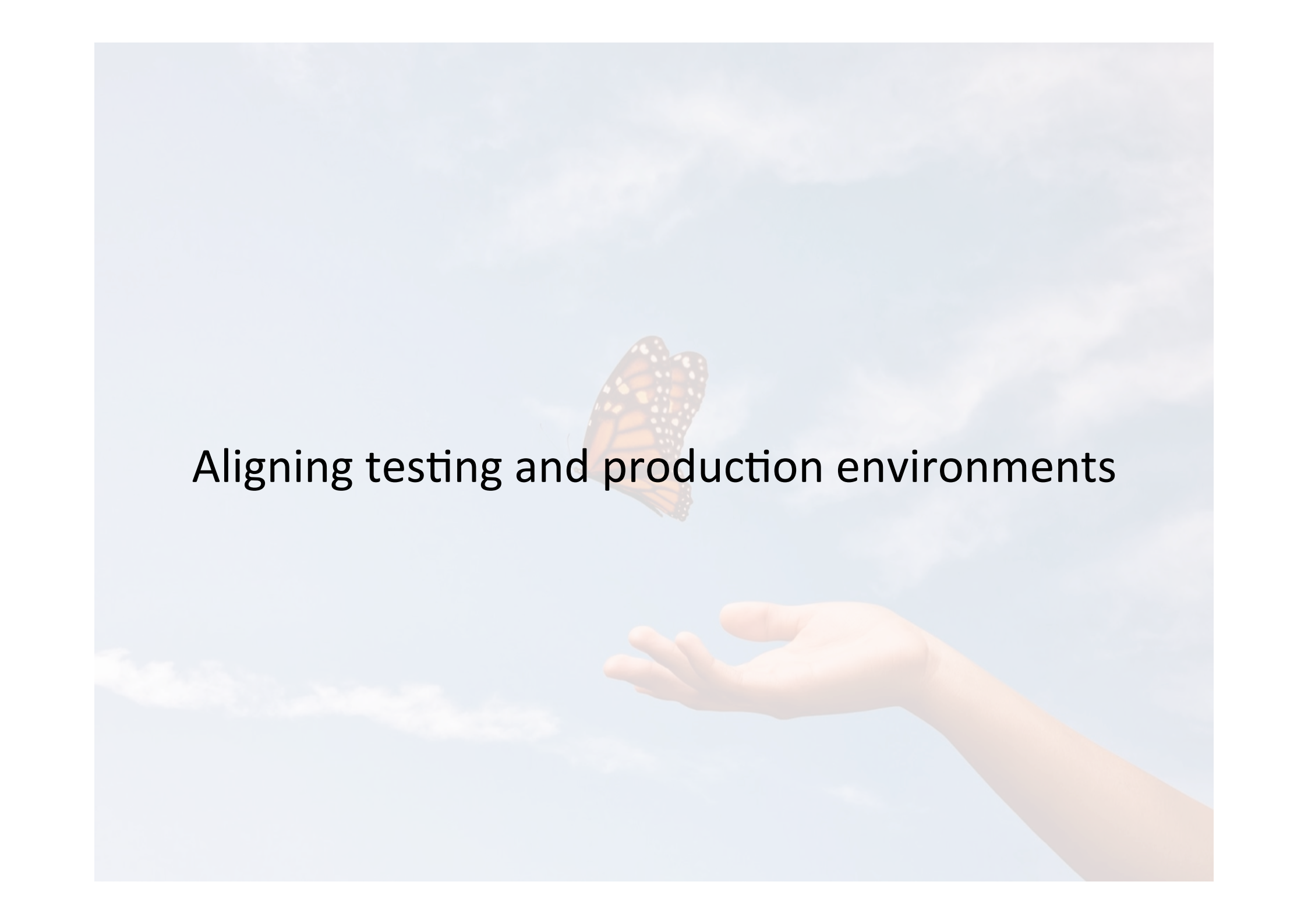
A conceptual image featuring a hand holding a monarch butterfly against a cloudy sky. The hand is positioned at the bottom right, with fingers slightly curled. The butterfly is in the center, with its wings spread, showing the characteristic orange and black pattern. The background is a light blue sky with soft, white clouds. The text "Deployment automation" is centered over the image.

Deployment automation

A hand holding a monarch butterfly against a cloudy sky. The hand is positioned at the bottom right, with the fingers slightly curled. The butterfly is perched on the hand, facing left. The background is a light blue sky with soft, white clouds.

According to Gartner, "Through 2015, 80% of outages impacting mission-critical services will be caused by people and process issues"

Source: Colville, R, and Spafford, G. Configuration Management for Virtual and Cloud Infrastructures

A photograph of a hand holding a monarch butterfly against a cloudy sky. The hand is positioned at the bottom right, with fingers slightly curled. The butterfly is in the center, with its wings spread, showing orange and black patterns. The background is a light blue sky with soft, white clouds. The text "Aligning testing and production environments" is overlaid in the center of the image.

Aligning testing and production environments

A photograph of a person's hand holding a monarch butterfly against a blue sky with white clouds. The text "Small batches" is overlaid on the image.

Small batches



Improved Productivity &
Efficiency

PRODUCTIVITY

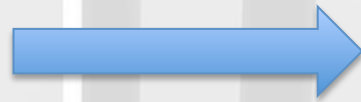


20%

Time saving on setting up and fixing test environments


PRODUCTIVITY

Days



Saved time on release: A few days' effort by operations engineers -> A click of a button by a developer

PRODUCTION



Saved effort (and frustration) on
troubleshooting and fixing issues caused by the
old release practice

PRODUCTIVITY

Collaborative culture and organization structure

AUTOMATE



ALL THE THINGS!

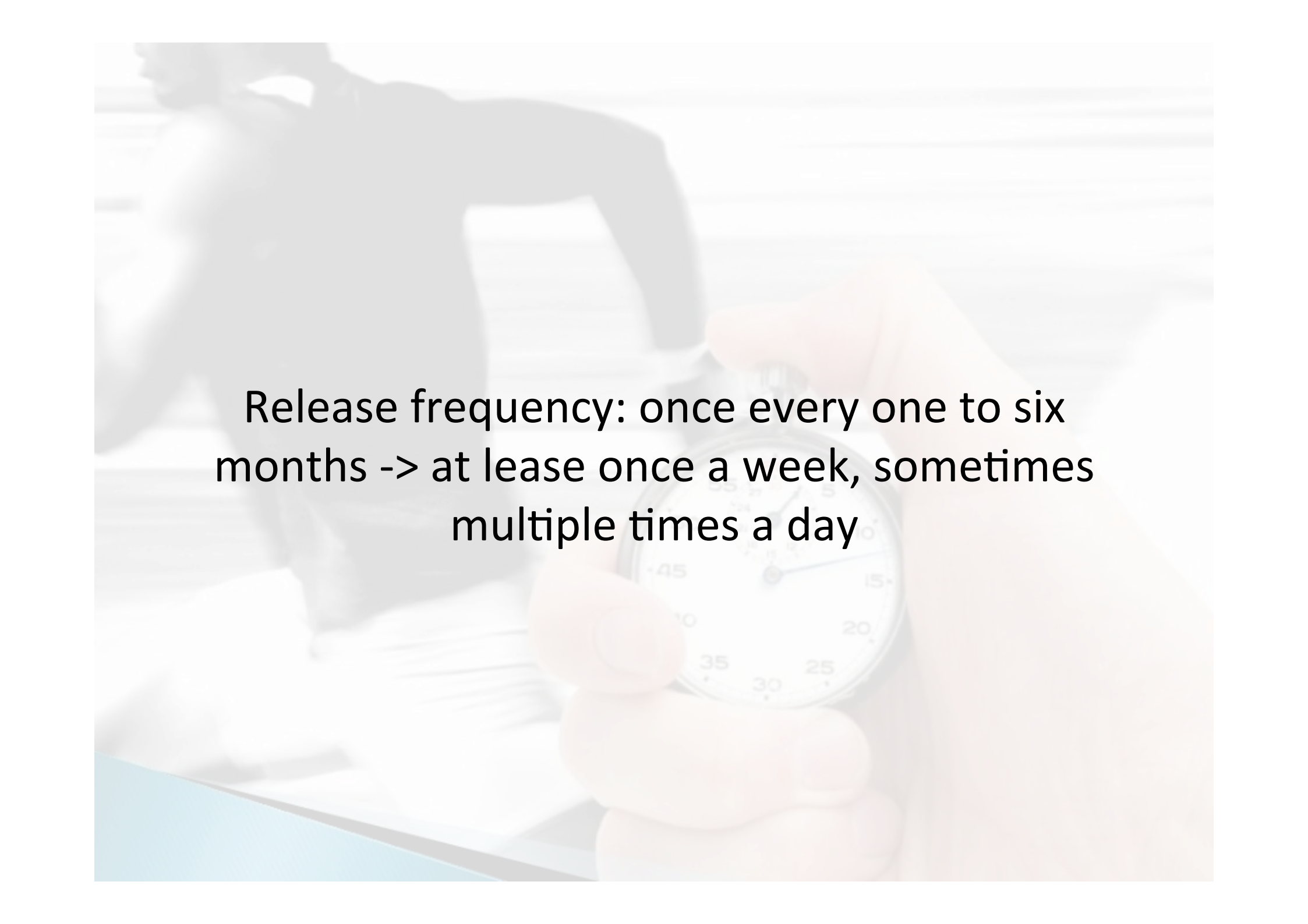


Aligning testing and production environments

PRODUCTIVITY

A hand holding a stopwatch in the foreground, with a blurred background of a person running on a track. The stopwatch is white with black markings and a red second hand. The background shows a person in a dark singlet and white shorts running on a track. The text "Accelerated Time to Market" is overlaid in the center.

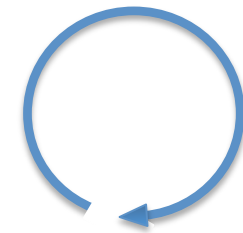
Accelerated Time to Market




Release frequency: once every one to six months -> at least once a week, sometimes multiple times a day



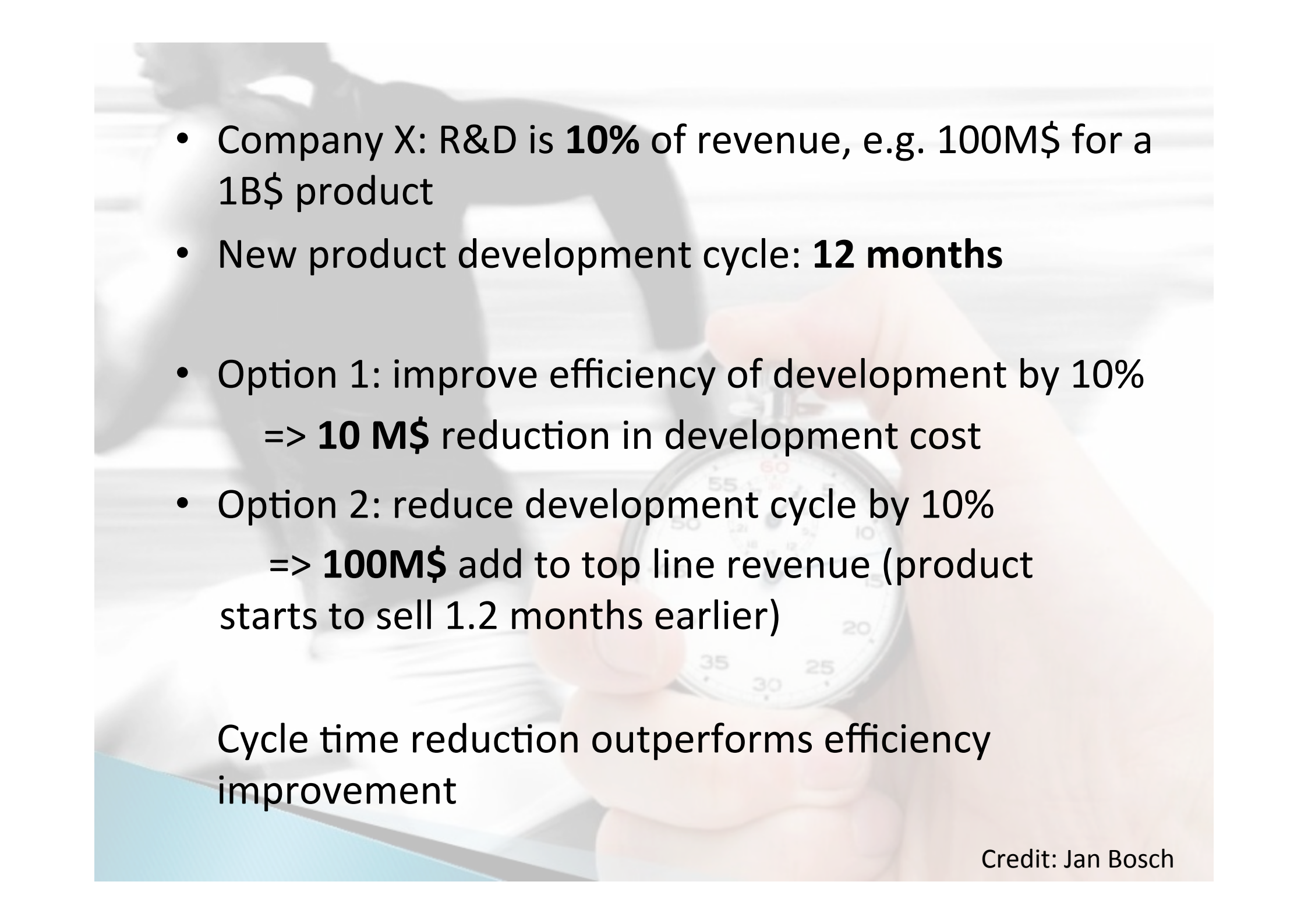
Cycle time: from a user story's conception to production



2-5 days



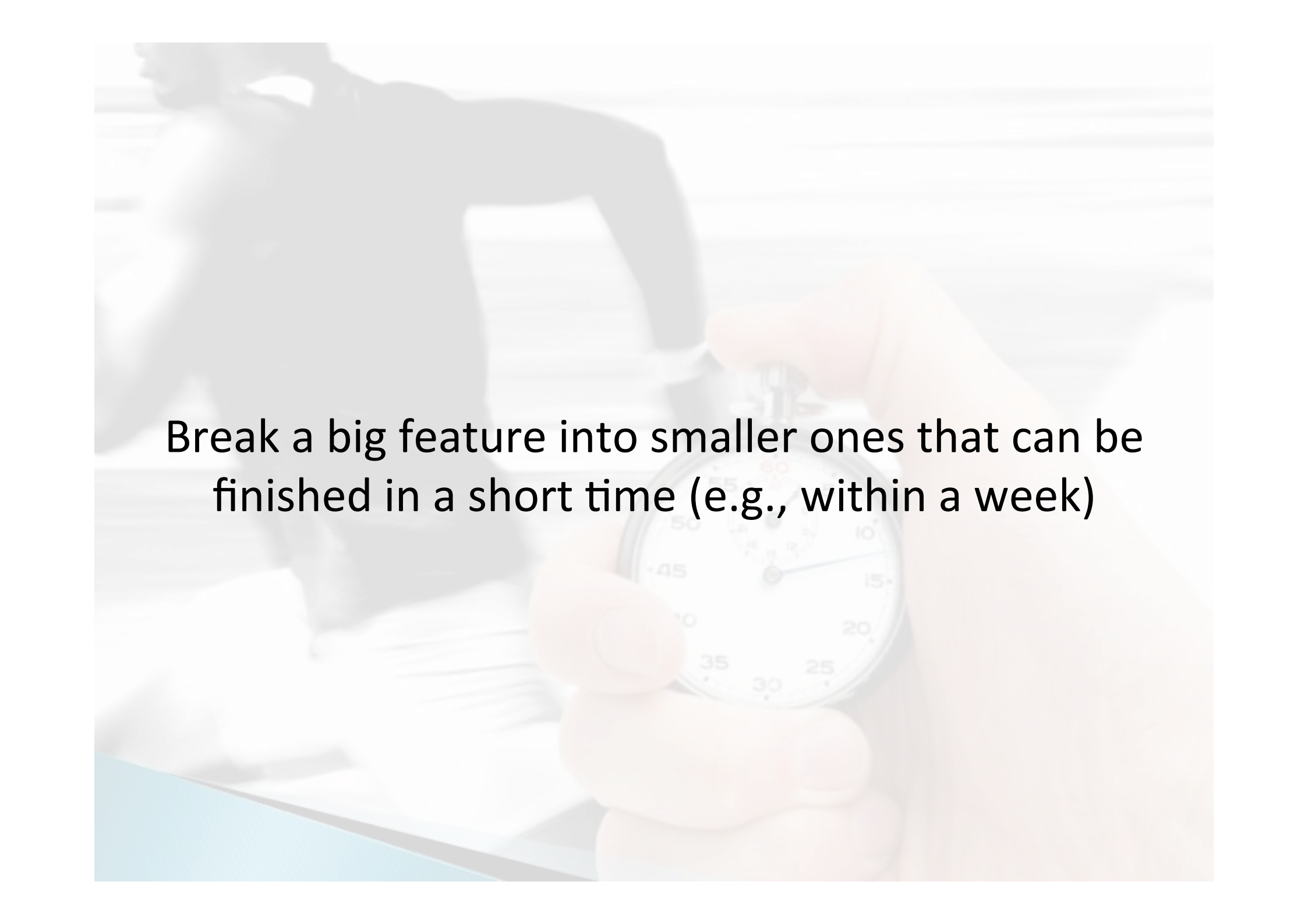
Deliver the business value inherent in new software releases to our customers more quickly

- 
- Company X: R&D is **10%** of revenue, e.g. 100M\$ for a 1B\$ product
 - New product development cycle: **12 months**
 - Option 1: improve efficiency of development by 10%
=> **10 M\$** reduction in development cost
 - Option 2: reduce development cycle by 10%
=> **100M\$** add to top line revenue (product starts to sell 1.2 months earlier)

Cycle time reduction outperforms efficiency improvement

A hand holding a silver stopwatch in the foreground, with a blurred background of a person running on a track. The stopwatch has a white face with black markings and numbers (5, 10, 15, 20, 25, 30, 35, 40, 45, 50) and a small sub-dial at the top. The text "Get fast feedback" is overlaid in the center of the image.


Get fast feedback




Break a big feature into smaller ones that can be finished in a short time (e.g., within a week)



Architecture: modifiability: allows adding features in small increments



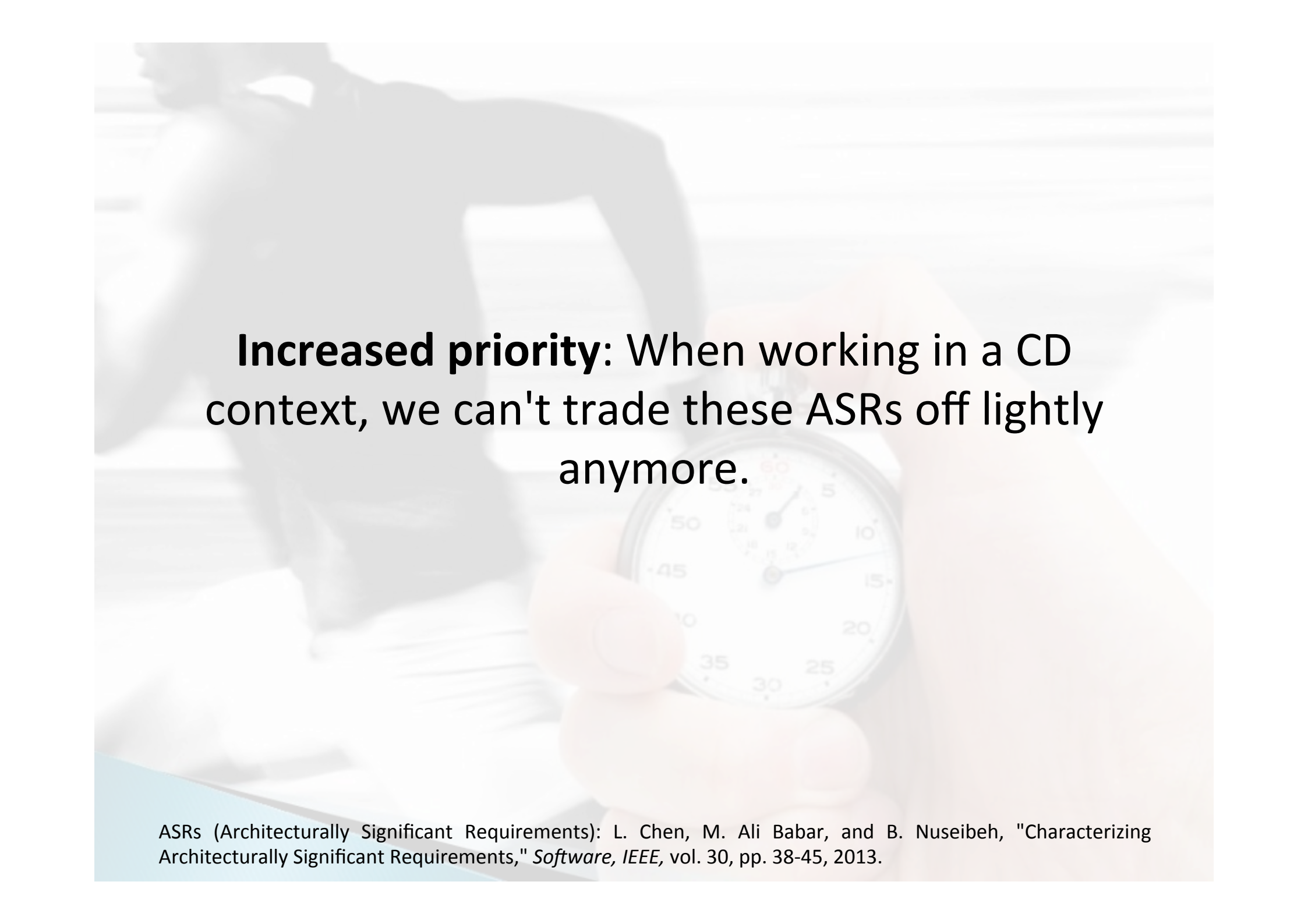
Architecture: deployability: a level of deployability that is acceptable for traditional multi-month release model may not be acceptable for CD



Architecture: testability: testing is not only about writing tests but also about design

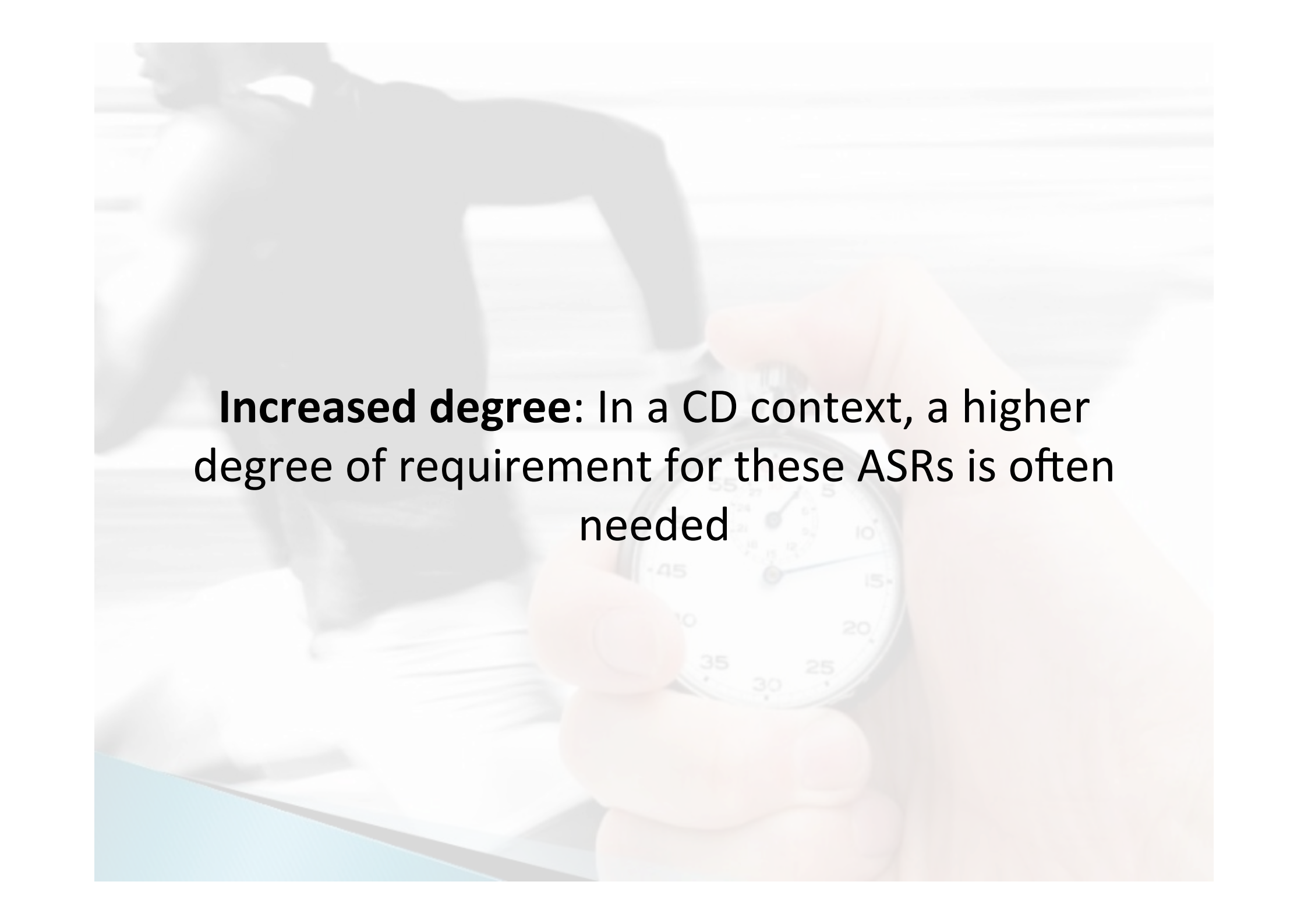


What's special compared to architecting in a traditional context?



Increased priority: When working in a CD context, we can't trade these ASRs off lightly anymore.

ASRs (Architecturally Significant Requirements): L. Chen, M. Ali Babar, and B. Nuseibeh, "Characterizing Architecturally Significant Requirements," *Software, IEEE*, vol. 30, pp. 38-45, 2013.



Increased degree: In a CD context, a higher degree of requirement for these ASRs is often needed

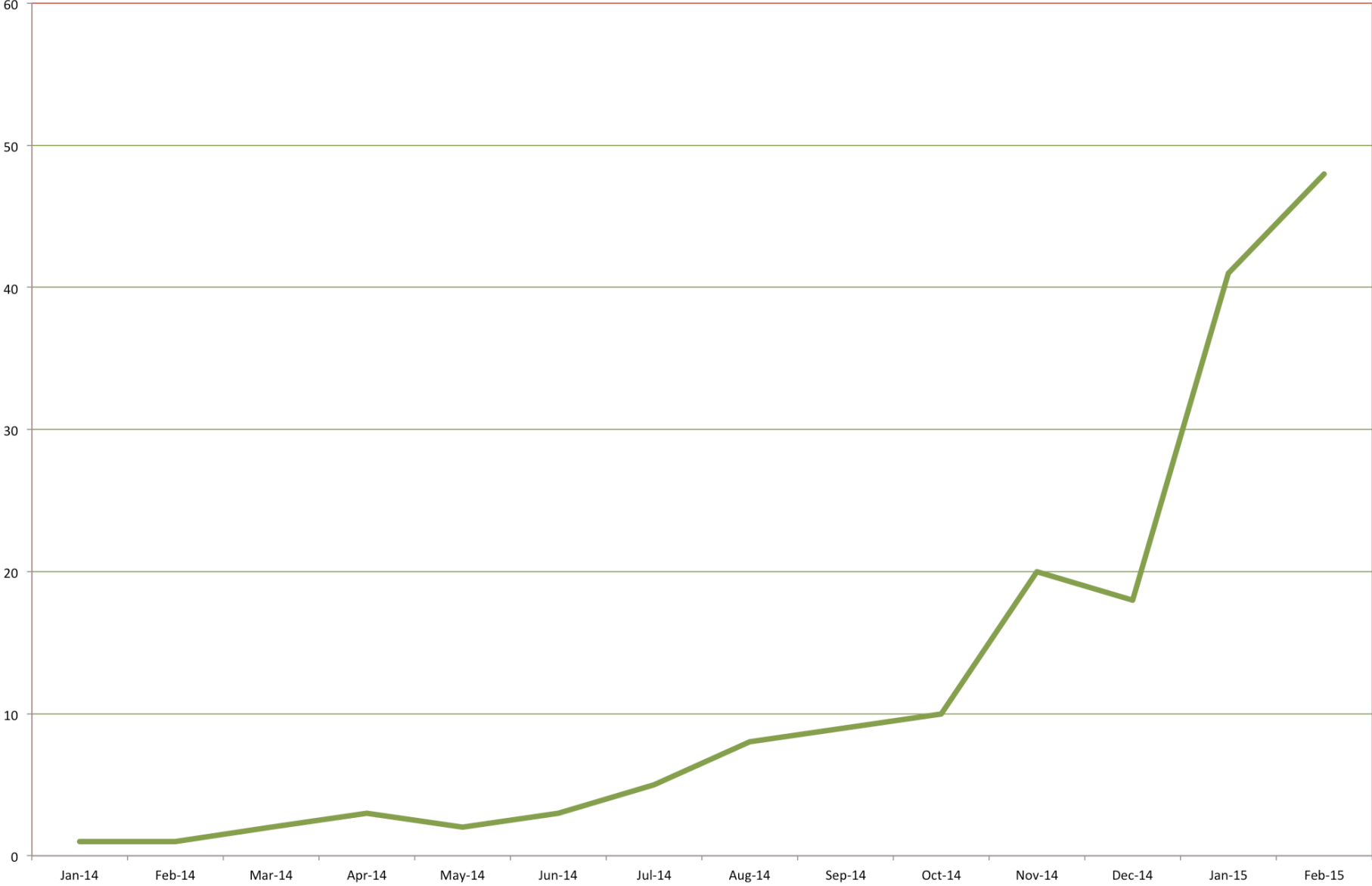
A hand holding a silver stopwatch in the foreground, with a blurred background of a person running on a track. The stopwatch has a white face with black markings and numbers. The background is a light, hazy scene of a person in motion on a track.

Automated tests

A hand holding a silver stopwatch in the foreground, with a blurred background of a person running on a track. The stopwatch has a white face with black markings and numbers. The background is a light blue and white track with a person in a dark singlet running. The overall image is semi-transparent.

Automated deployment, a click of a button

releases per-month



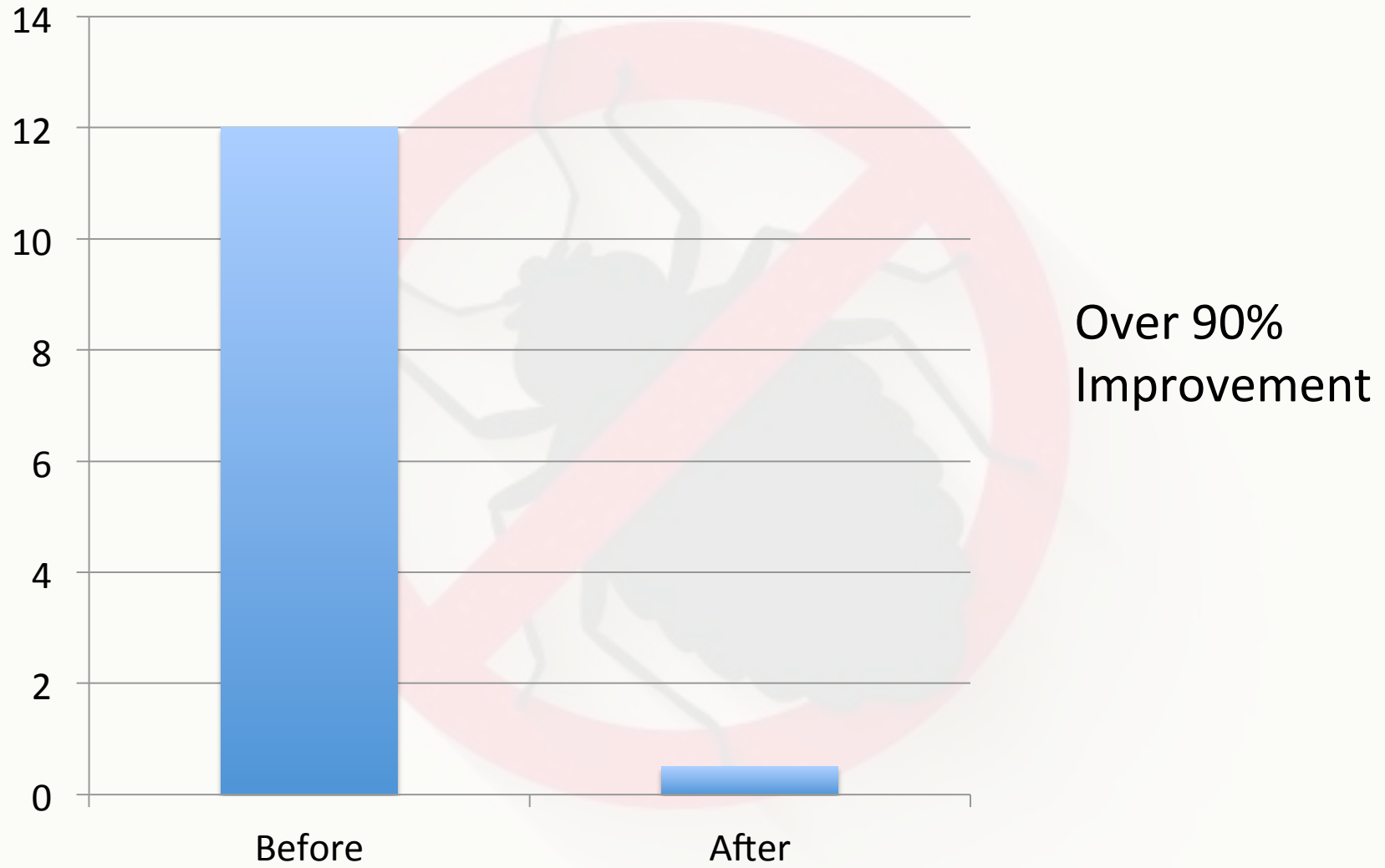
A hand holding a silver stopwatch in the foreground, with a blurred background of a person running on a track. The stopwatch has a white face with black markings and numbers. The text "Optimize process" is overlaid in the center.

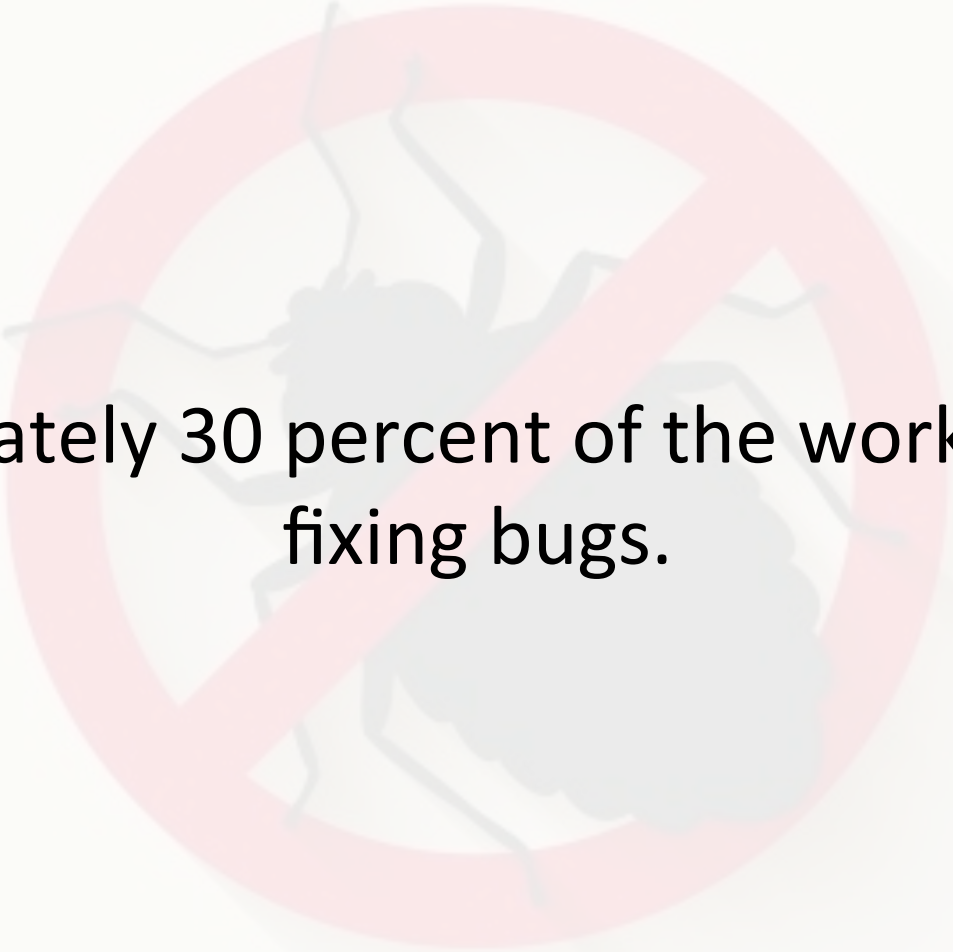
Optimize process



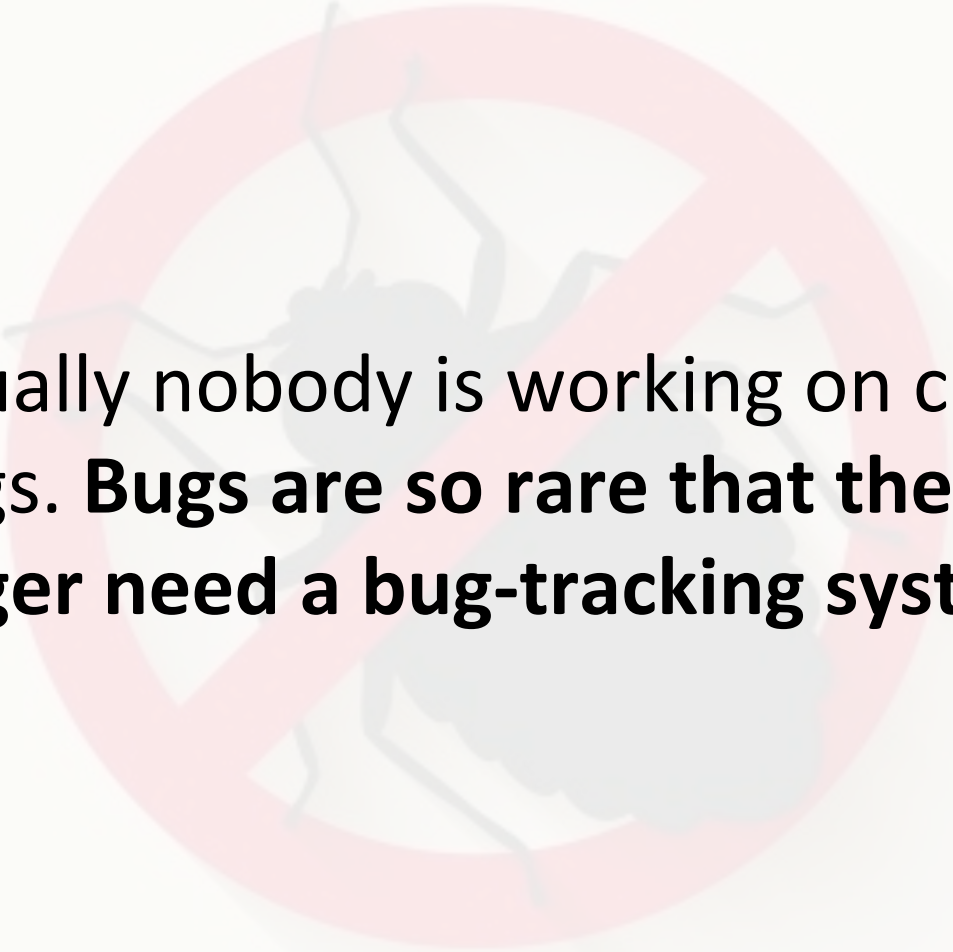
Improved Product Quality

Customer Found Bugs

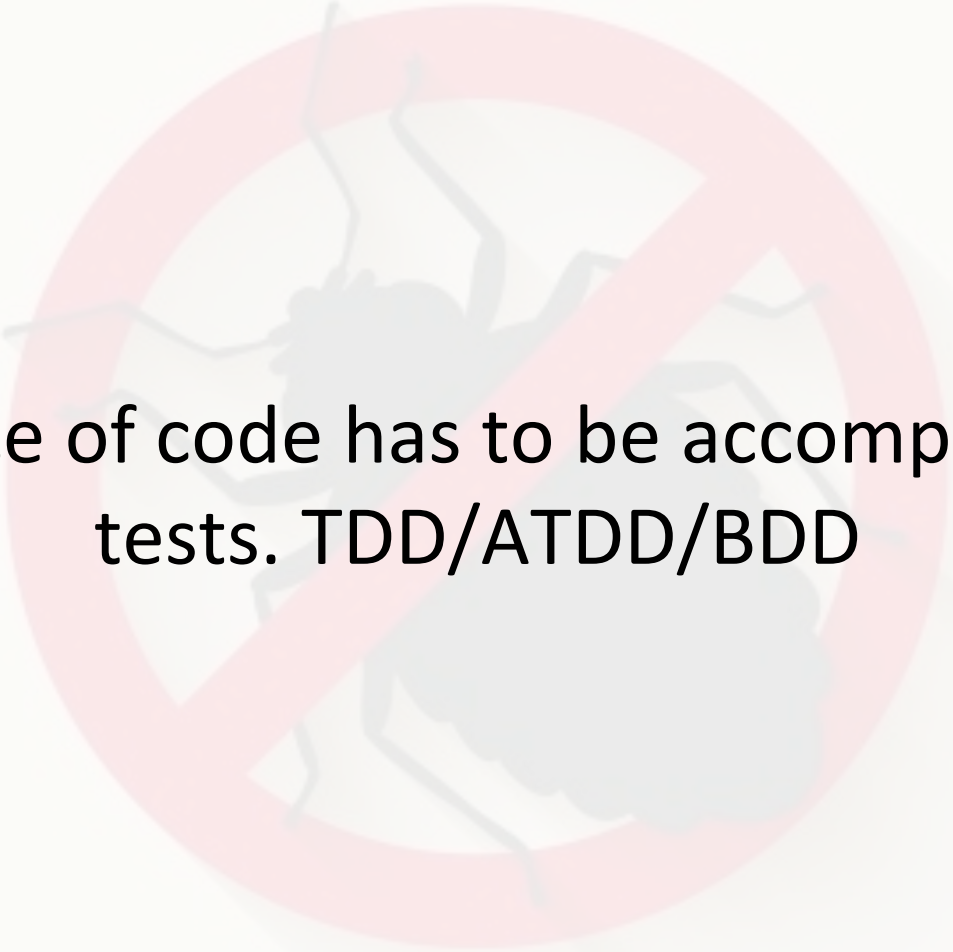





Approximately 30 percent of the workforce was fixing bugs.



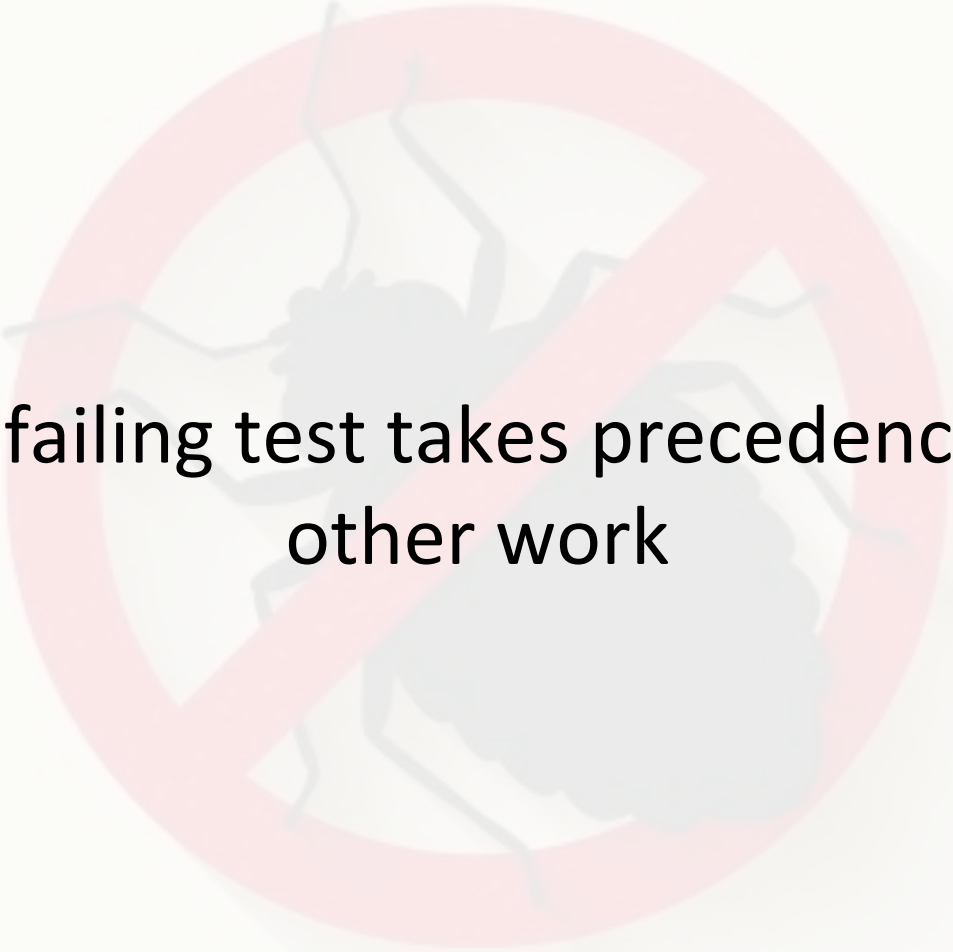
Now, usually nobody is working on customer-found bugs. **Bugs are so rare that the teams no longer need a bug-tracking system.**



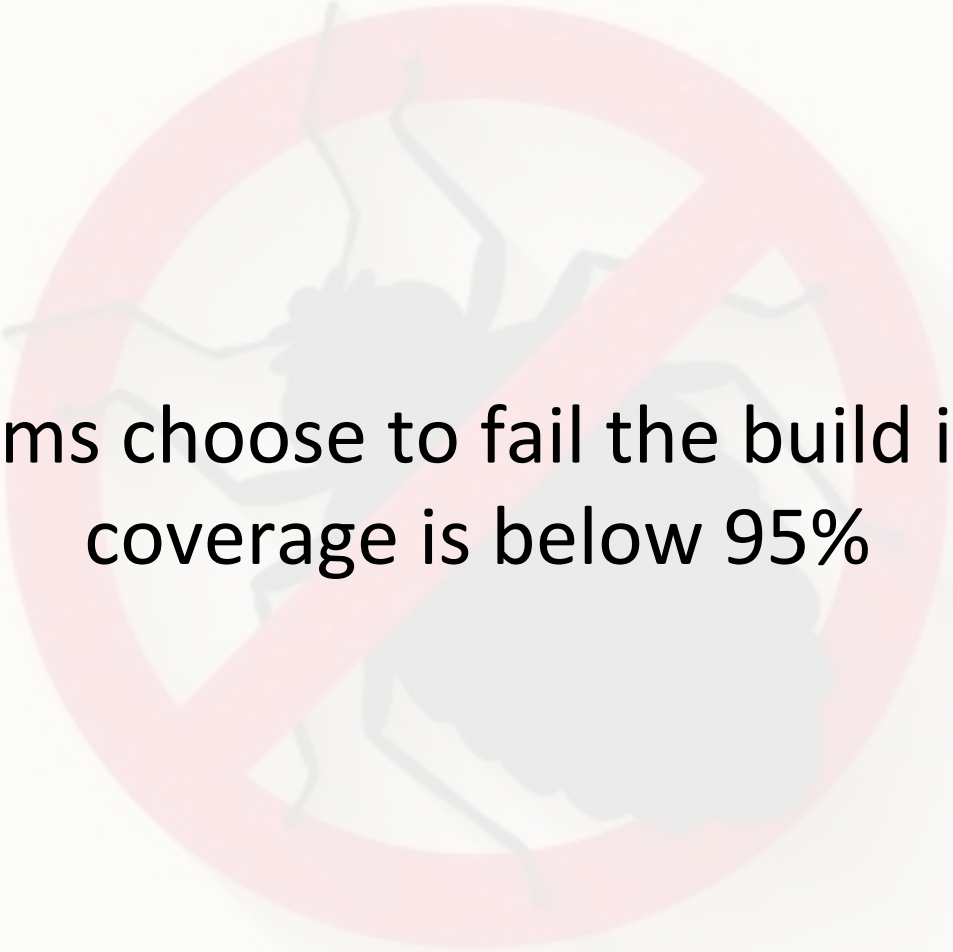
Any piece of code has to be accompanied by tests. TDD/ATDD/BDD



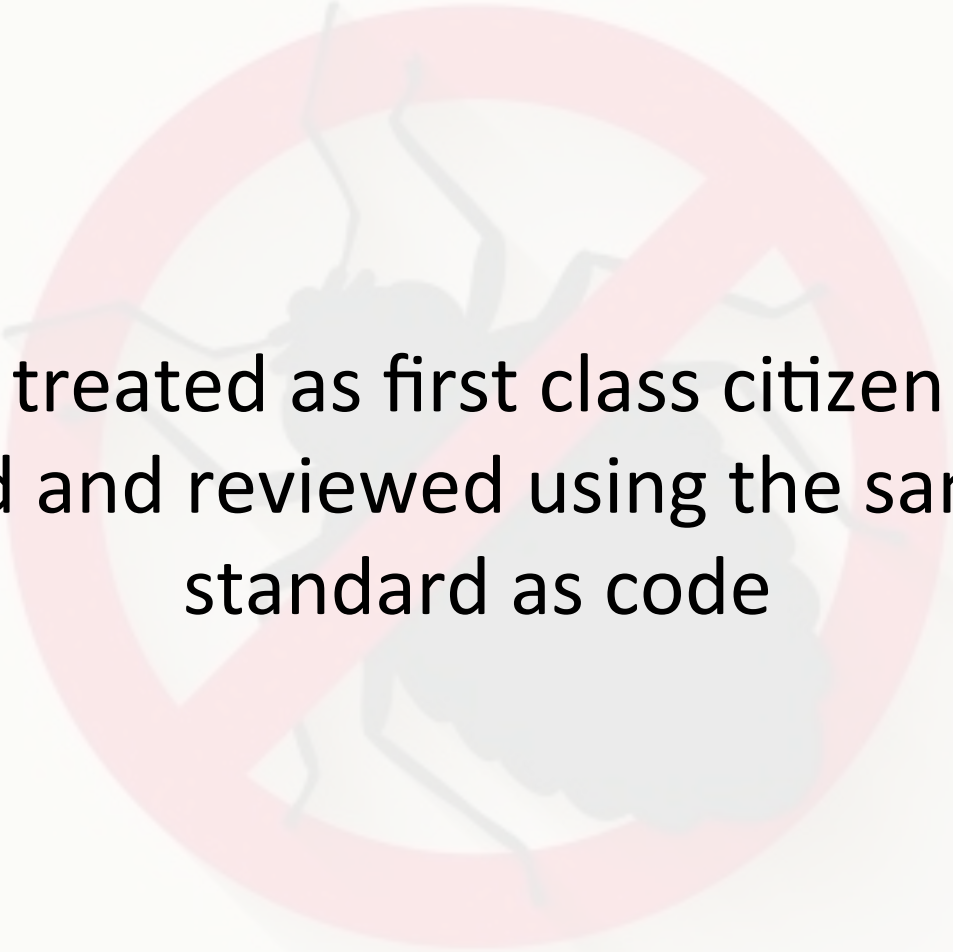
Eliminate flaky (non-deterministic) tests



Fixing any failing test takes precedence over any other work



Most teams choose to fail the build if the test coverage is below 95%



Tests are treated as first class citizen: they are developed and reviewed using the same quality standard as code

Main Dashboard

Time changes... [Configure widgets](#)

Unit Tests Coverage

97.4%

Line Coverage **97.7%**

Condition Coverage **96.2%**

Debt	0	Issues	0
! Blocker	0		
+ Critical	0		
+ Major	0		
✓ Minor	0		
✓ Info	0		

Complexity

838

/Function	2.4	/Class	7.8
/File	7.8		

Functions Files

Lines Of Code	Files	
3,858	108	
Java	Directories	Lines
	22	4,923

Functions

348

Classes	Statements
108	1,506

Accessors **90**

Duplications

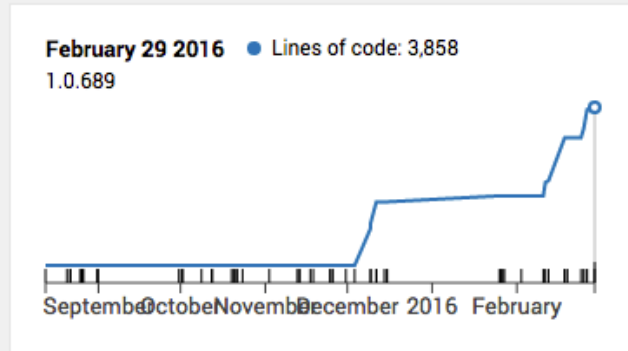
0.0%

Lines	Blocks	Files
0	0	0

Most Violated Rules

Any severity [More](#)

No result



Main Dashboard

Time changes... [dropdown arrow]

Unit Tests Coverage **99.9%**

Line Coverage **99.9%**

Condition Coverage **100.0%**

Unit Test Success **100.0%**

Failures	Errors	Tests
0	0	671

Execution Time **11.1 sec** [arrow]

Debt **0**

Issues **0**

Blocker	0
Critical	0
Major	0
Minor	0
Info	0

Complexity **435**

/Function	/Class
1.5	5.0

/File **5.1**

Legend: Functions Files

Lines Of Code **3,195**

Files **86**

Java

Directories	Lines
120	4,394

Functions **272**

Classes	Statements
87	727

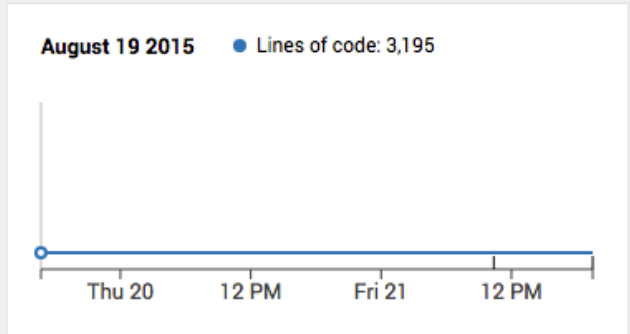
Accessors **2**

Duplications **0.0%**

Lines	Blocks	Files
0	0	0

Most Violated Rules **Any severity** [dropdown arrow] [More](#)

No result



Main Dashboard

Time changes... [Configure widgets](#)

Unit Tests Coverage

98.7%

Line Coverage

98.9%

Condition Coverage

97.5%

Debt

0

- ! Blocker 0
- + Critical 0
- + Major 0
- ✓ Minor 0
- ✓ Info 0

Issues

0

Complexity

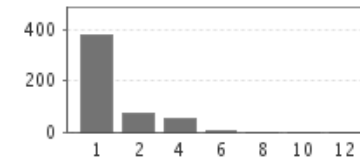
860

/Function /Class

1.6 3.6

/File

3.7



Functions Files

Lines Of Code

6,481

Java

Files

232

Directories

154

Lines

8,579

Functions

537

Classes

238

Statements

1,441

Accessors

6

Duplications

0.0%

Lines

0

Blocks

0

Files

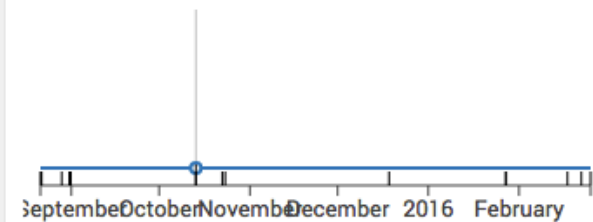
0

Most Violated Rules Any severity

[More](#)

No result

October 13 2015 ● Lines of code: 6,481
1.0.478



Main Dashboard

Time changes...

Unit Tests Coverage
95.0%

Line Coverage
95.7%

Condition Coverage
89.9%

Debt
1d 6h

Issues
16

- Blocker 0
- Critical 11
- Major 5
- Minor 0
- Info 0

Complexity
2,120

/Function 1.9 /Class 4.8

/File
4.9

Functions Files

Lines Of Code
17,985

Files
436

Java Directories 112 Lines 23,460

Functions
1,106

Classes 442 Statements 5,485

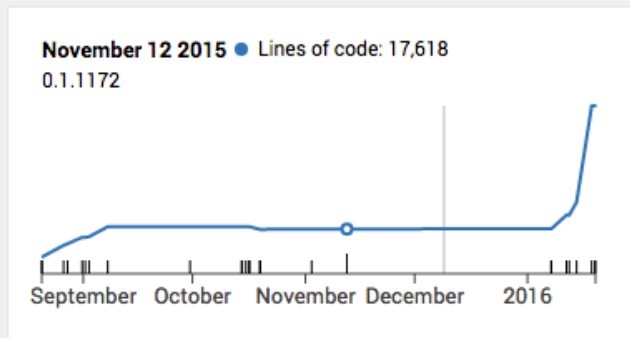
Accessors
1,140

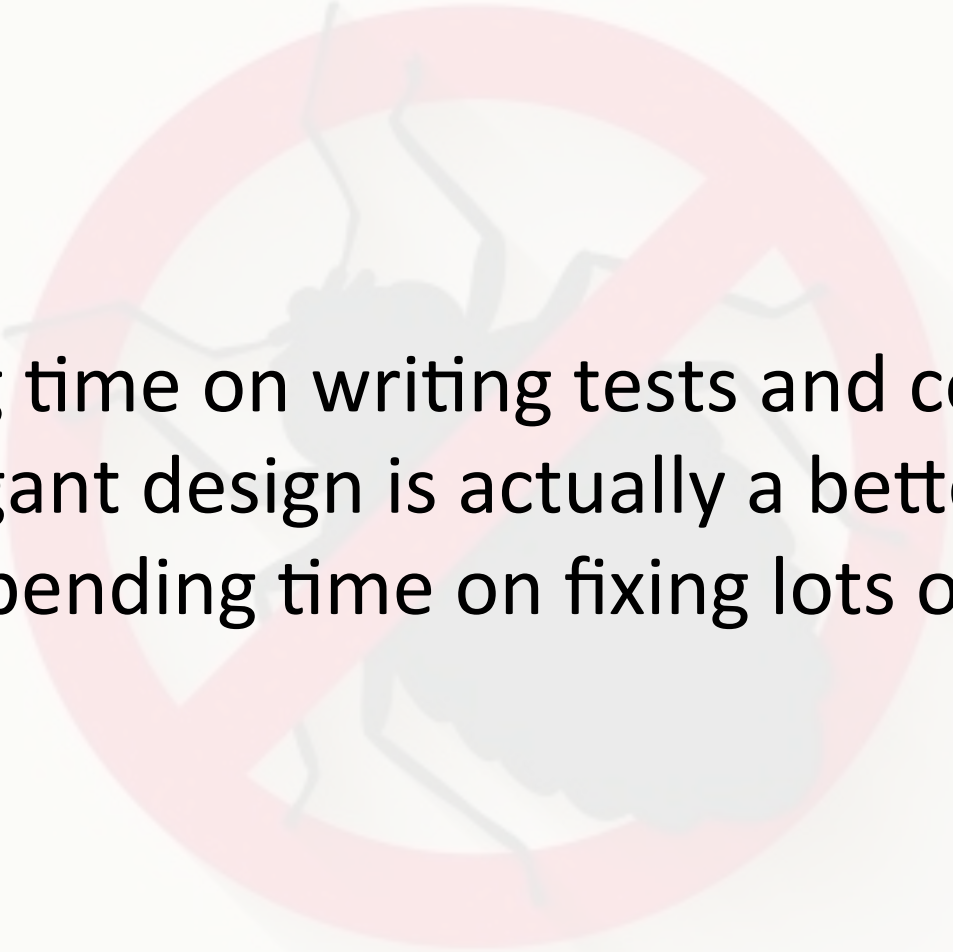
Duplications
2.3%

Lines 530 Blocks 35 Files 24

Most Violated Rules Any severity [More](#)

- Method Length 11
- Anon Inner Length 2
- Reliance on default encoding 1
- Ncss Method Count 1
- Unused Private Field 1





Spending time on writing tests and coming up with elegant design is actually a better option than spending time on fixing lots of bugs.

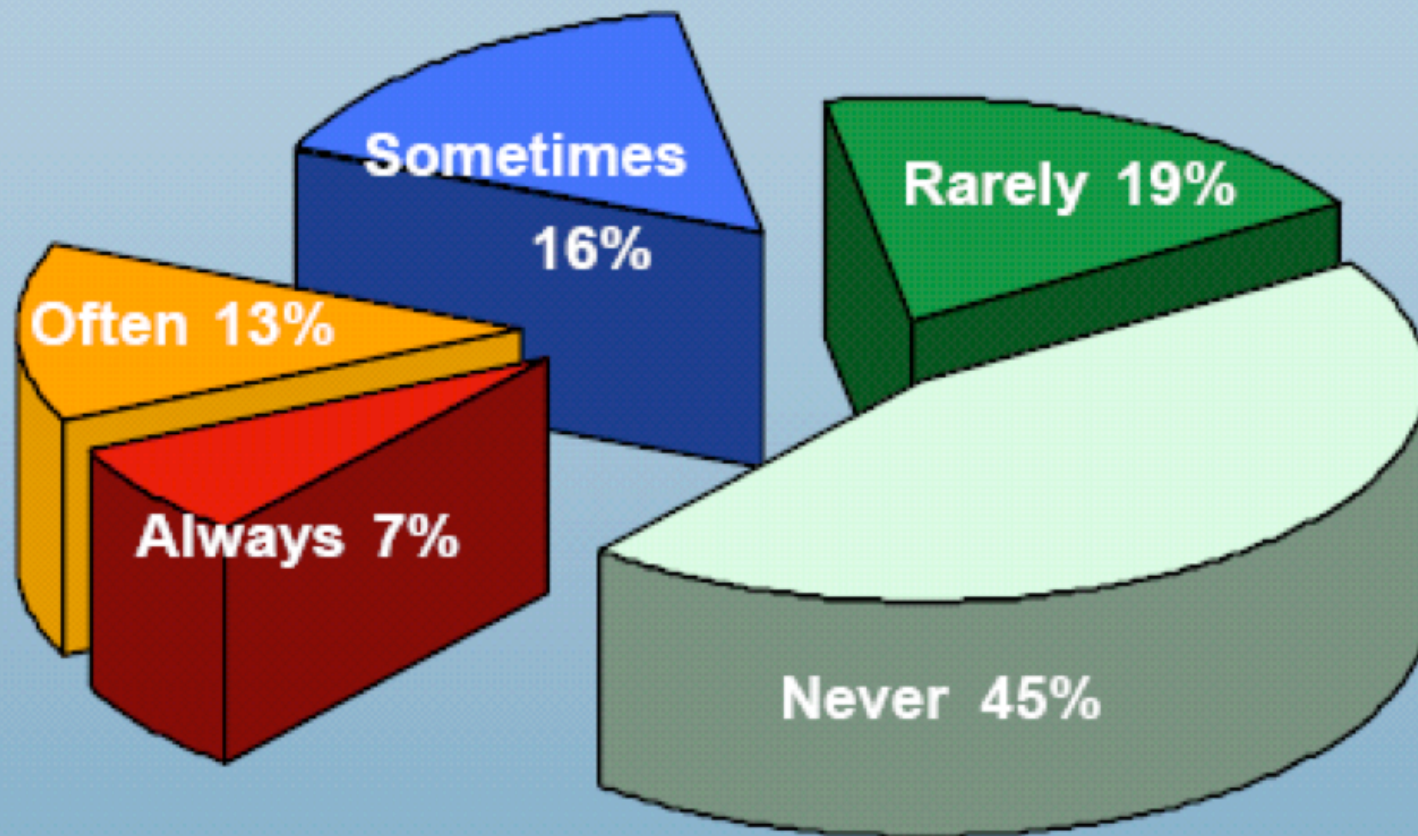
The background of the slide is a blurred image of a target and an arrow. The target is on the left, and the arrow is on the right, pointing towards the target. The image is heavily blurred, creating a sense of motion and focus. The colors are muted, with a mix of greens, yellows, and browns.

Building the Right Product

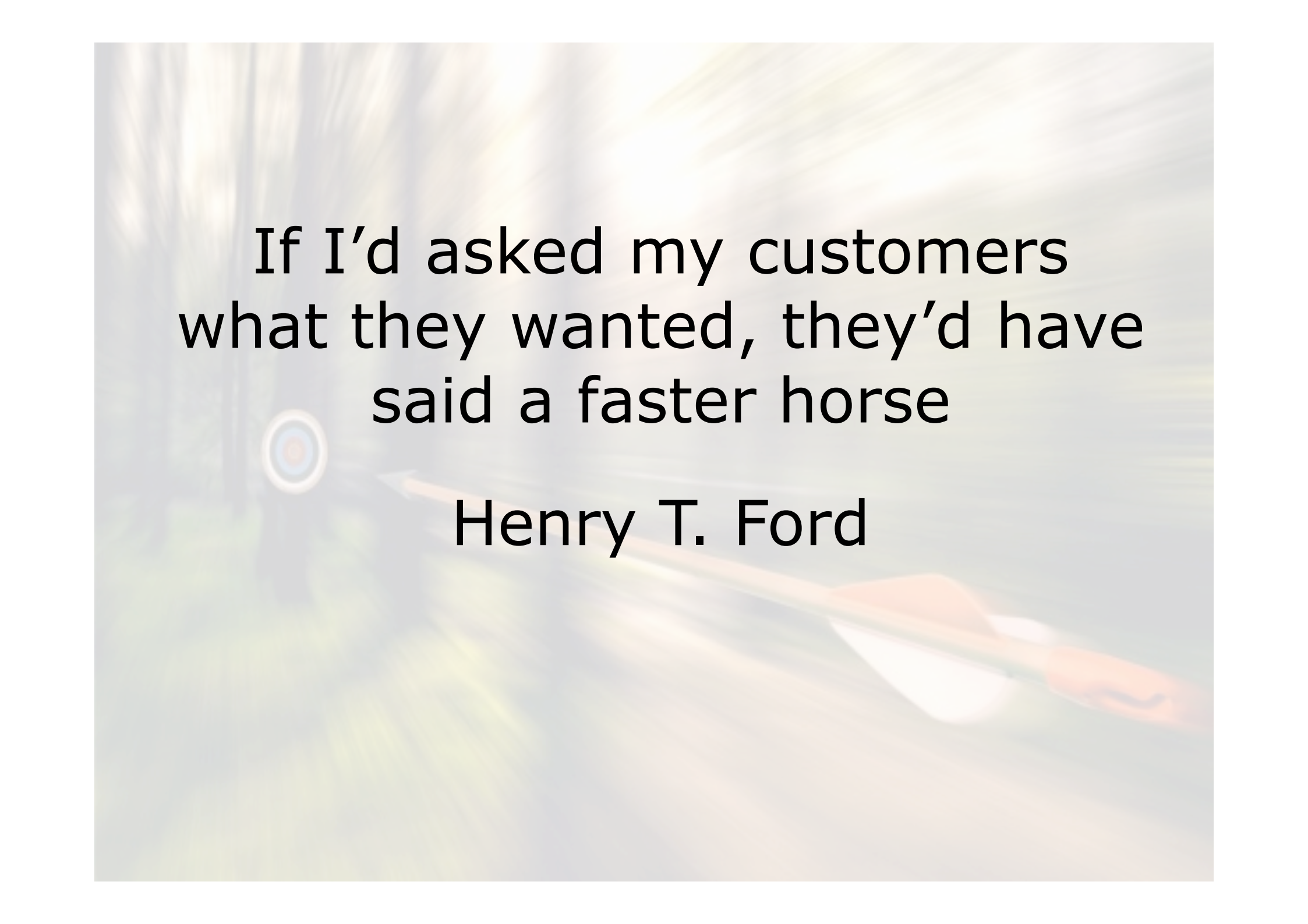
Features / Functions Used in a Typical System

**Often / Always
Used: 20%**

**Rarely / Never
Used: 64%**

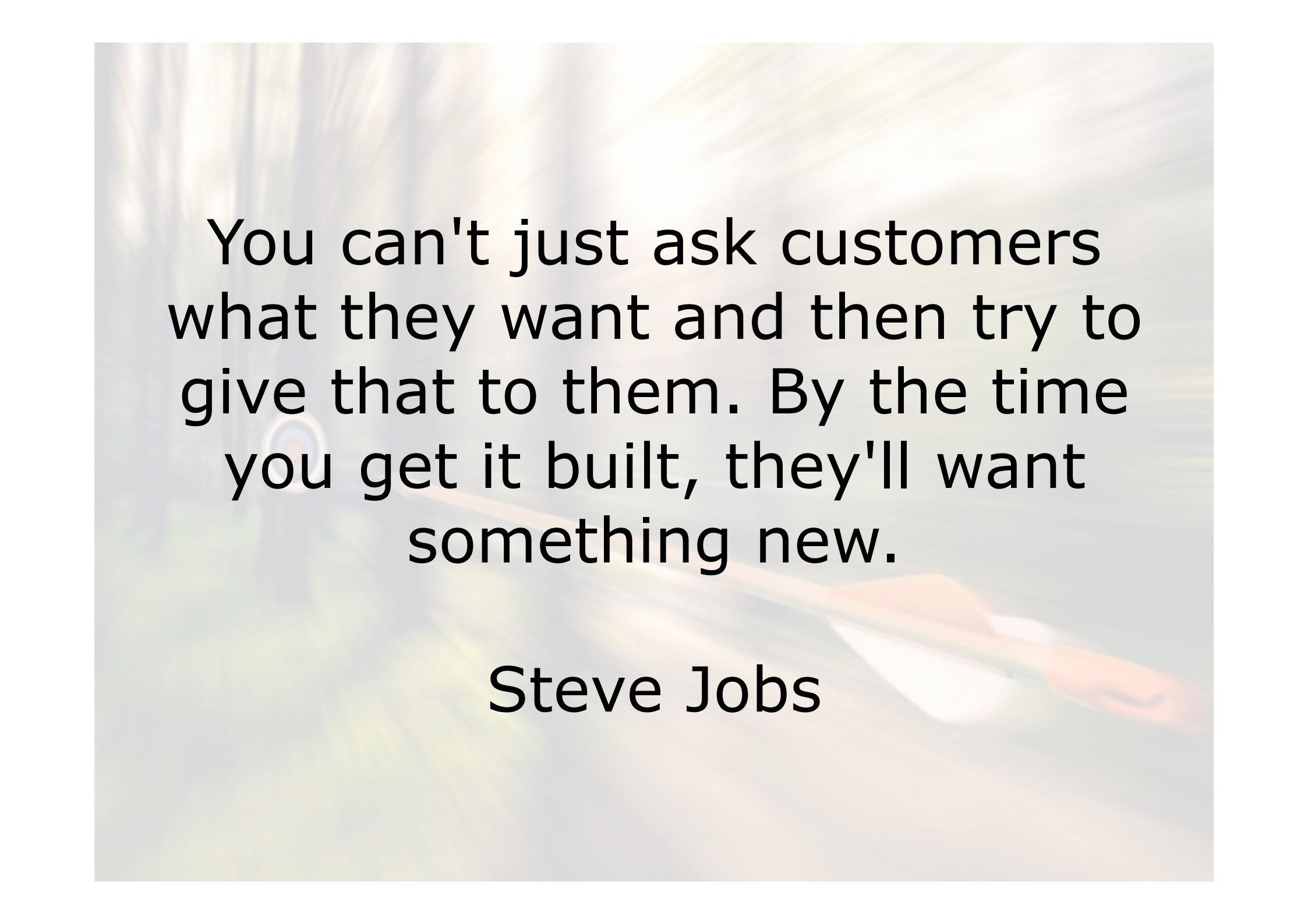


Standish Group Study Reported at XP2002 by Jim Johnson, Chairman

The background of the slide is a blurred image of an archery target. An arrow is shown in motion, having just hit the bullseye of the target. The target has concentric rings of blue, red, and yellow. The arrow is orange and white, with a white fletching. The background is a mix of light and dark colors, suggesting an outdoor setting with trees and a bright sky.

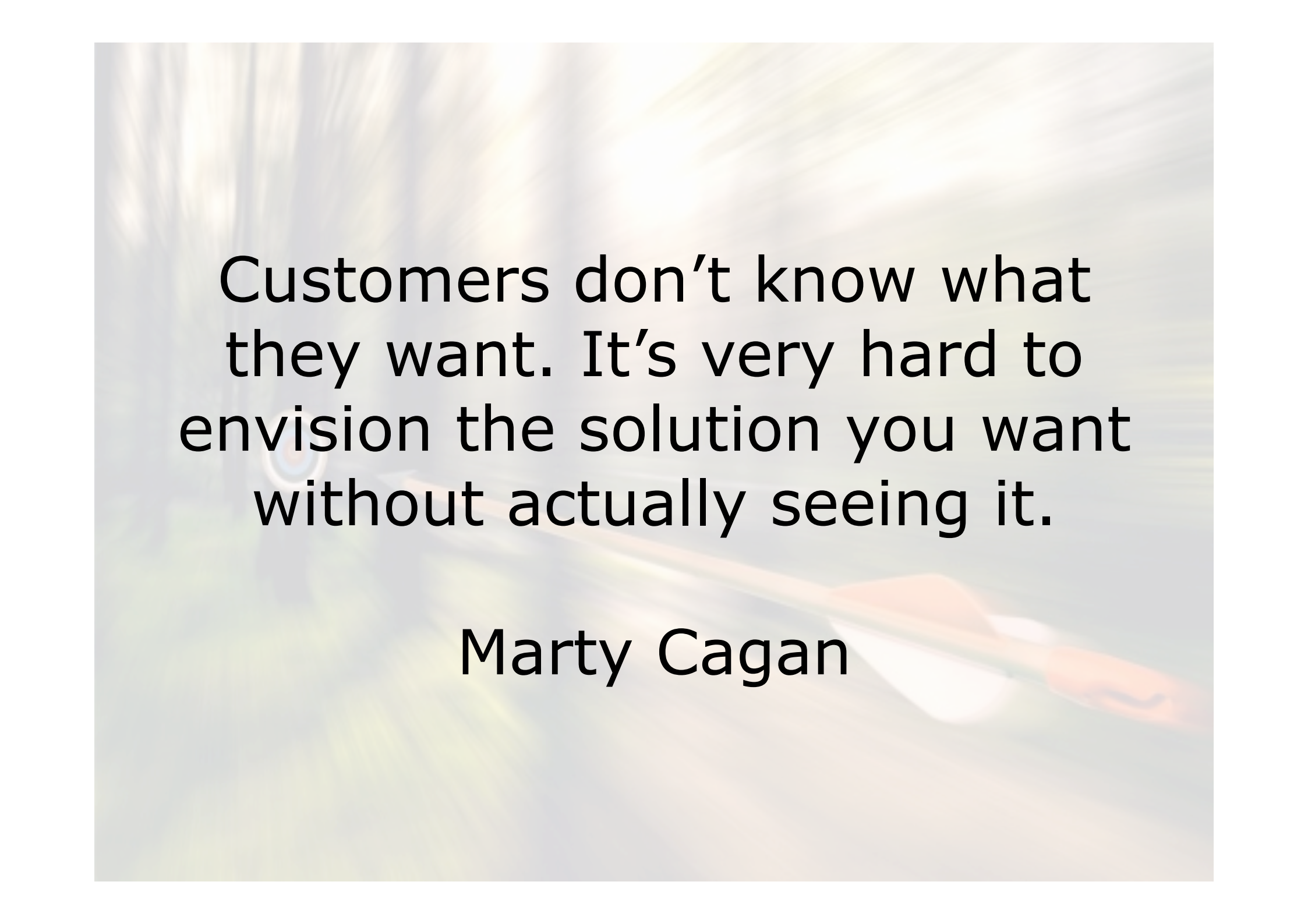
If I'd asked my customers
what they wanted, they'd have
said a faster horse

Henry T. Ford

A blurred background image showing a hand holding a pen over a document, suggesting a business or creative context.

You can't just ask customers what they want and then try to give that to them. By the time you get it built, they'll want something new.

Steve Jobs



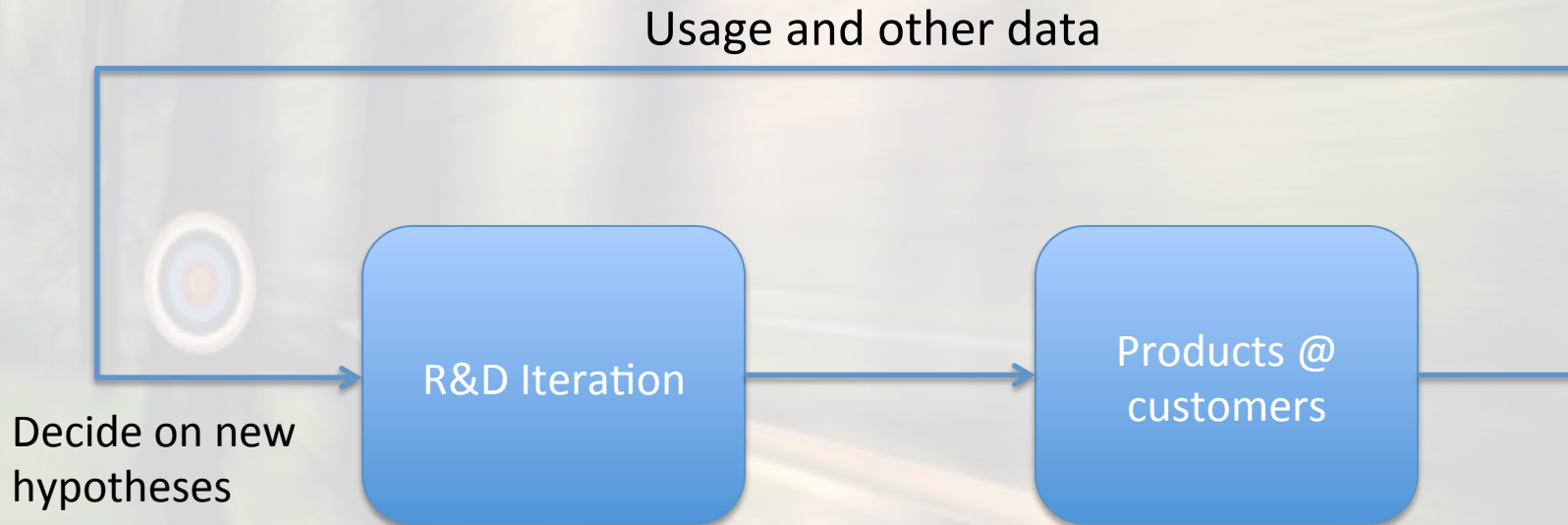
Customers don't know what they want. It's very hard to envision the solution you want without actually seeing it.

Marty Cagan

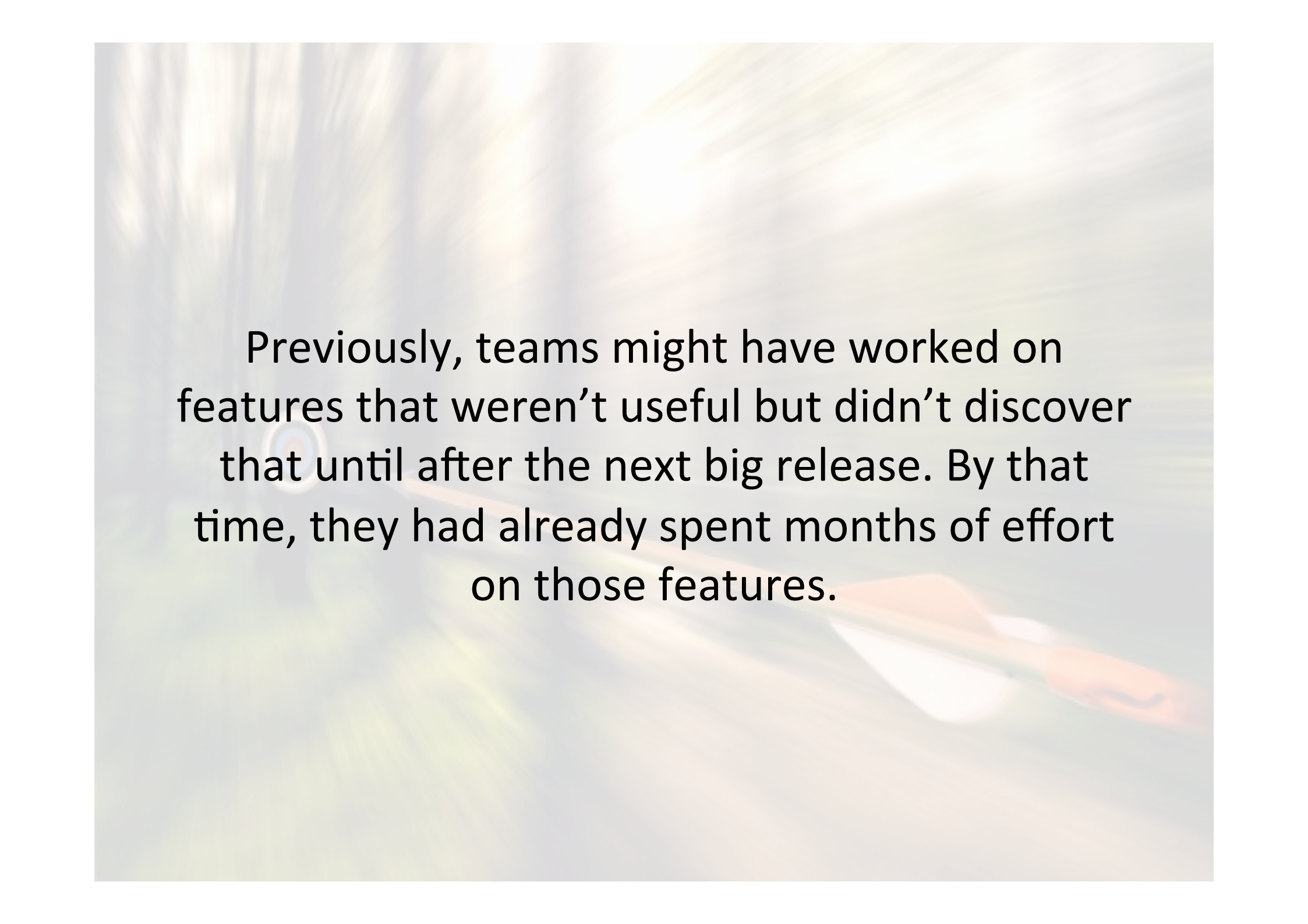
A blurred background image of a forest with a wooden arrow pointing towards the text.

Not Requirements; Hypotheses

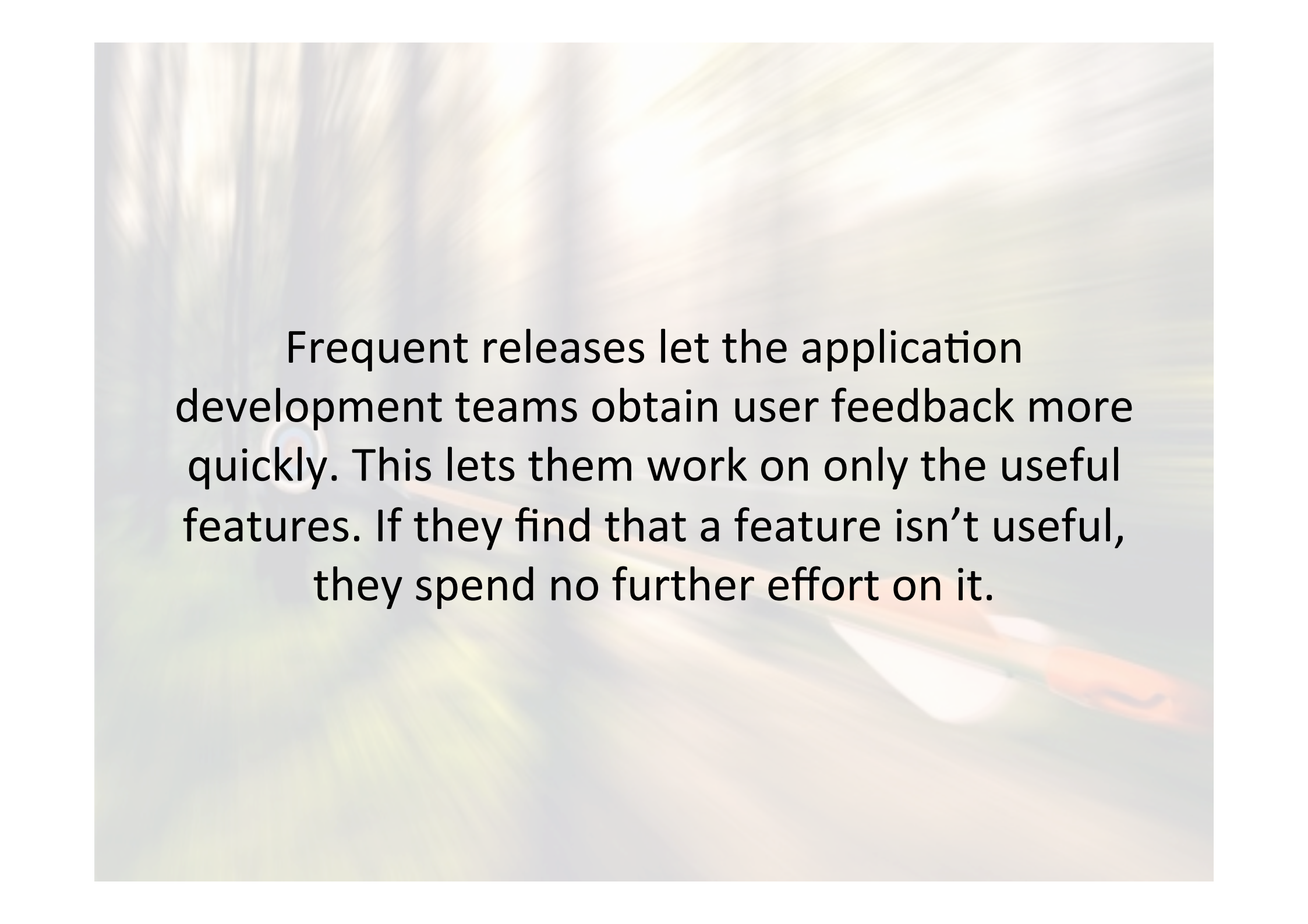
The company running the most experiments against the lowest cost per experiment wins



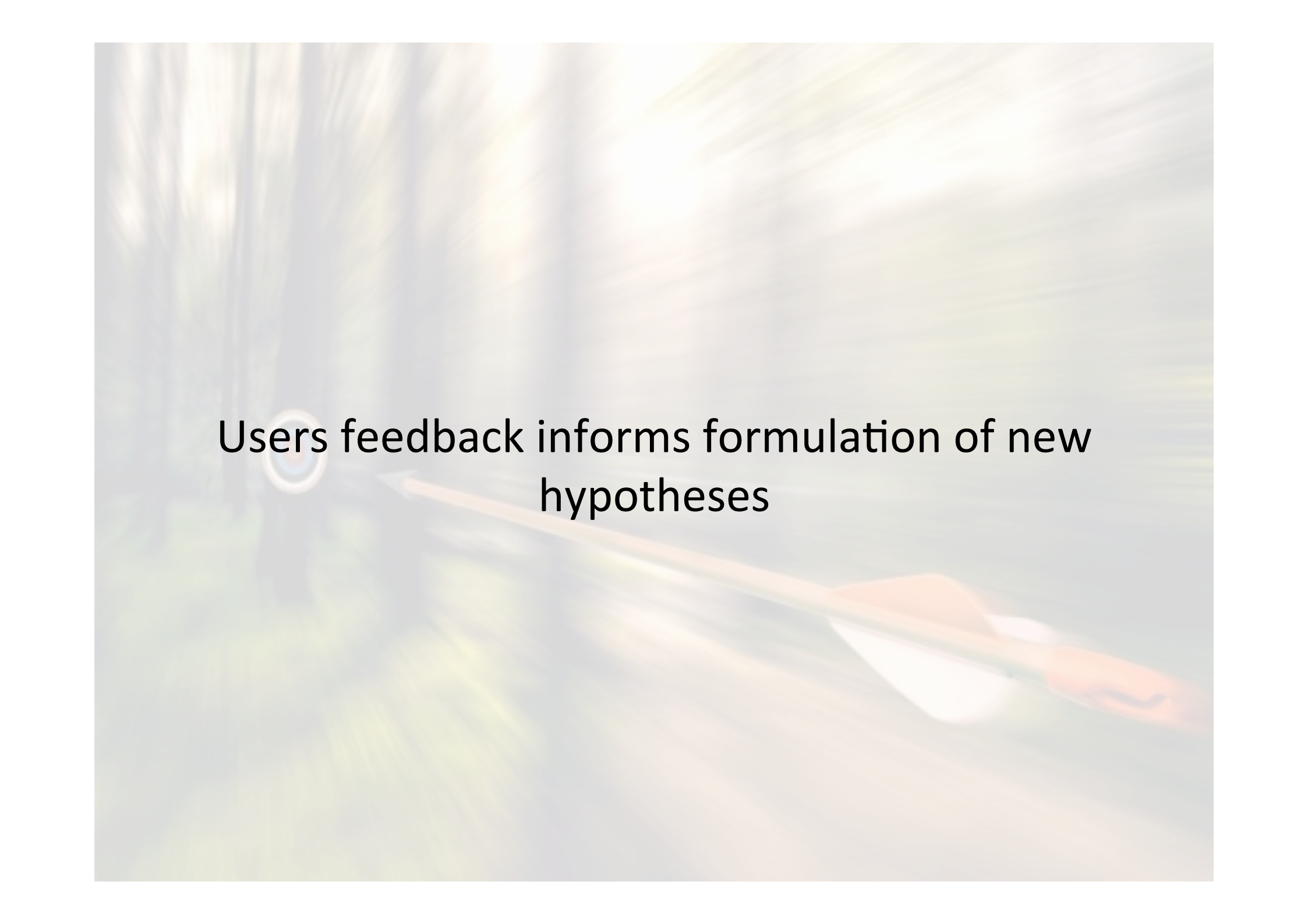
Decisions should be based on DATA, not opinions

A blurred background image showing a hand holding a pen over a document, suggesting a writing or editing process. The text is centered over this background.

Previously, teams might have worked on features that weren't useful but didn't discover that until after the next big release. By that time, they had already spent months of effort on those features.

A background image showing a hand holding a pen over a document. A circular highlight is visible on the text. The text is centered and reads: Frequent releases let the application development teams obtain user feedback more quickly. This lets them work on only the useful features. If they find that a feature isn't useful, they spend no further effort on it.

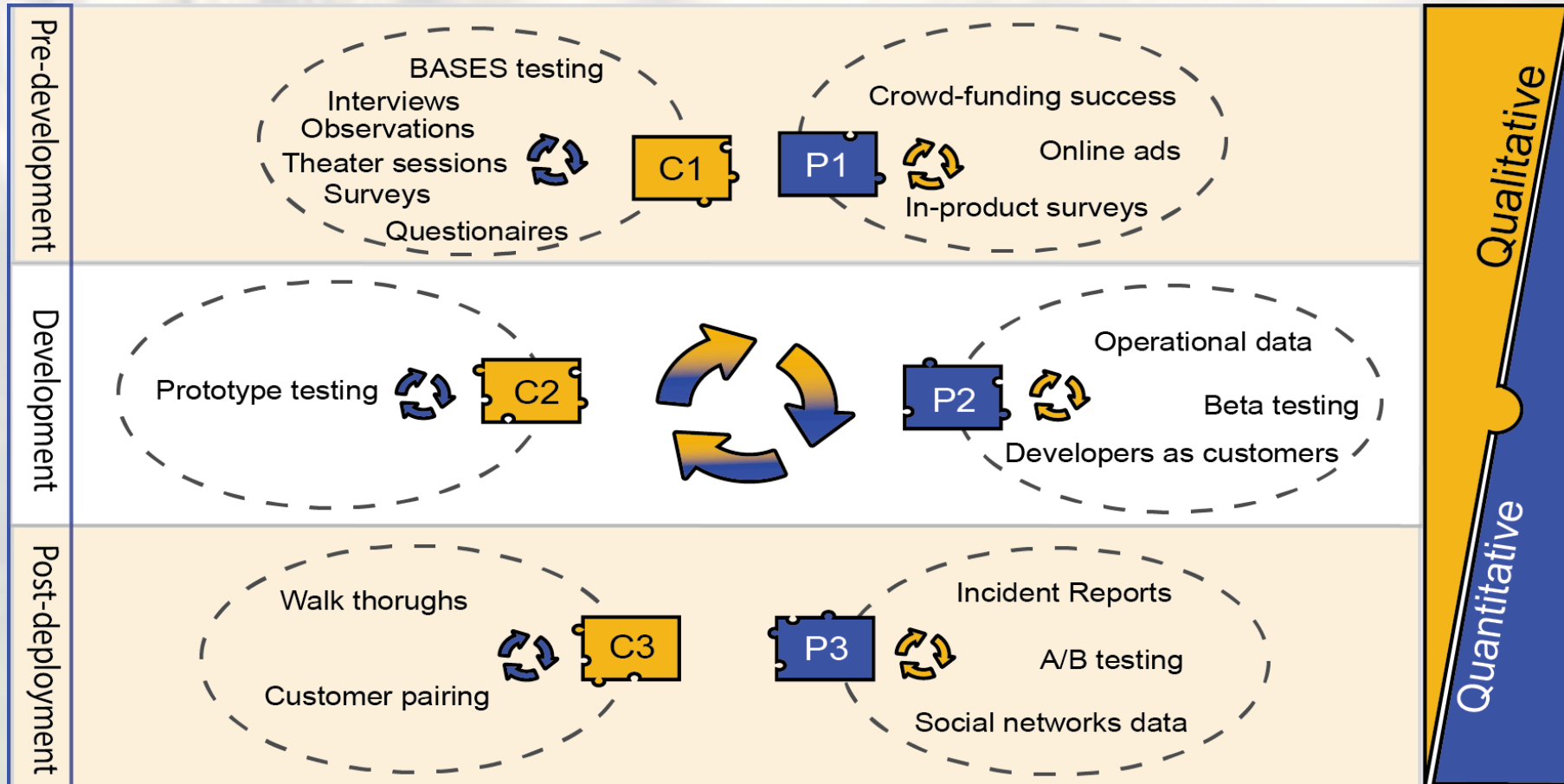
Frequent releases let the application development teams obtain user feedback more quickly. This lets them work on only the useful features. If they find that a feature isn't useful, they spend no further effort on it.

The background of the slide is a blurred photograph of an archery target. An arrow is visible, having just struck the center bullseye. The image is out of focus, with vertical lines suggesting a camera panning or a fast shutter speed. The colors are muted, with greens, browns, and greys.

Users feedback informs formulation of new hypotheses

The background is a blurred photograph of a forest. In the center-left, there is a target with a bullseye. An arrow is shown in motion, having just hit the target, with a long, bright orange and white trail behind it. The text "Fast and continuous gathering of user feedback" is overlaid on the image.

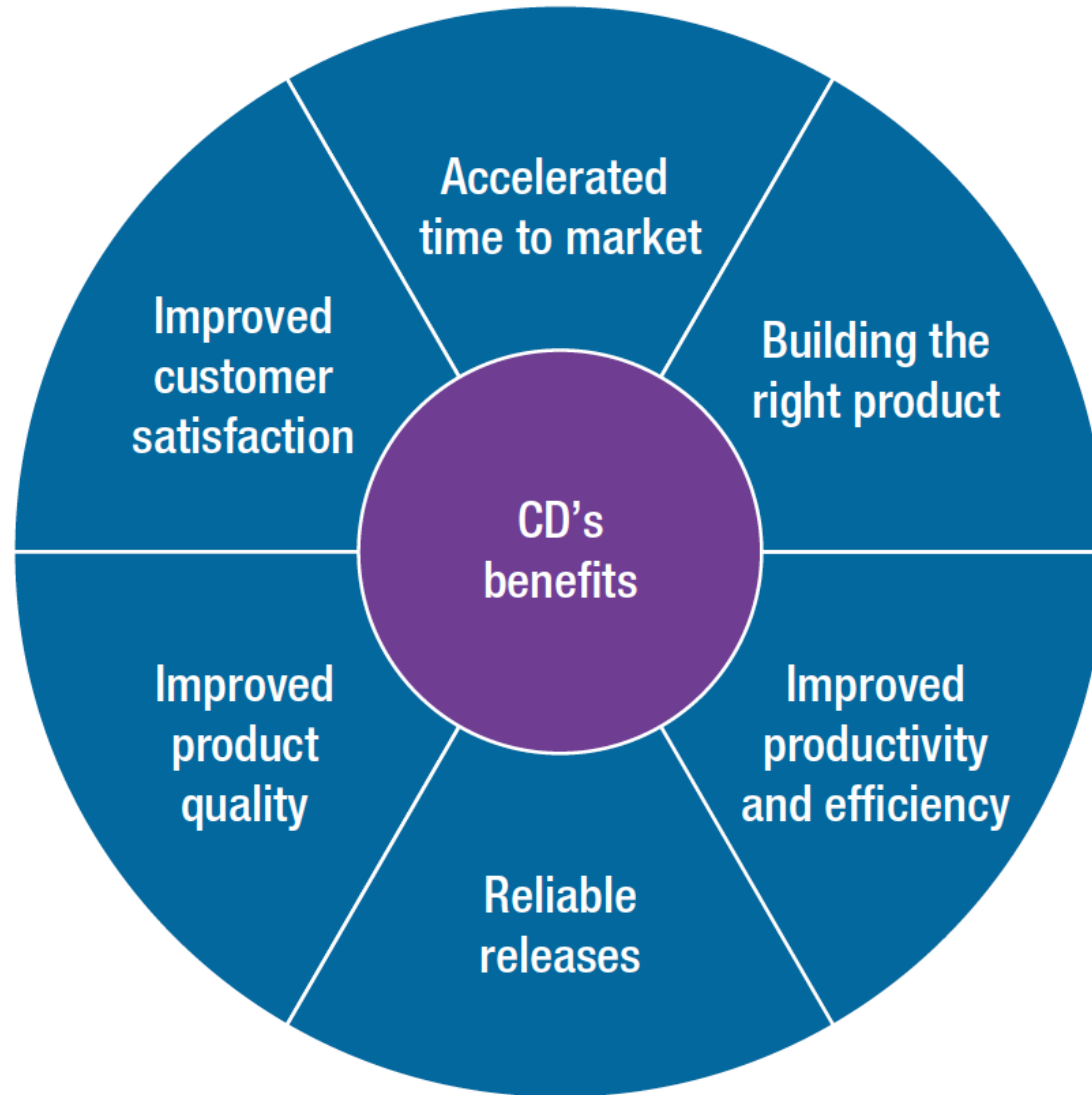
Fast and continuous gathering of user feedback



Source: A. Fabijan, H. H. Olsson, and J. Bosch, "Customer Feedback and Data Collection Techniques in Software R&D: A Literature Review," in *Software Business: 6th International Conference, ICSOB 2015, Braga, Portugal, June 10-12, 2015*.

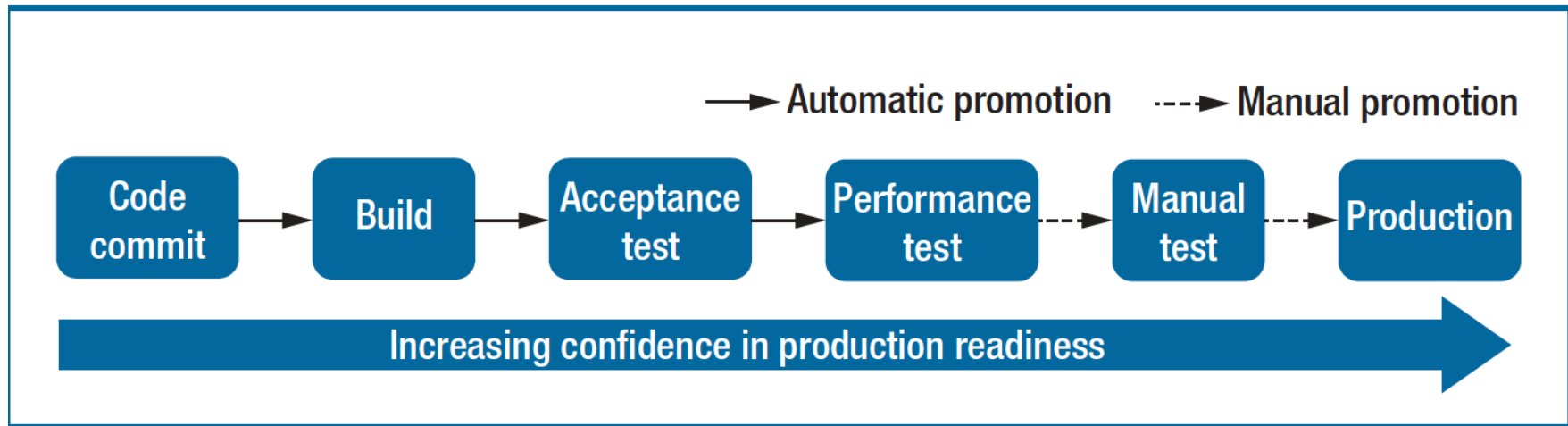
Improved Customer Satisfaction





Source: L. Chen, "Continuous Delivery: Huge Benefits, but Challenges Too," IEEE Software, vol. 32, 2015.

However, to achieve/maximize these benefits,
we need to seriously take CD as a **holistic
approach.**



An example CD pipeline. (Source: L. Chen, "Continuous Delivery: Huge Benefits, but Challenges Too," IEEE Software, vol. 32, 2015.)

1. **Requirements** as hypotheses
2. Break big feature into smaller ones than can be finished in a short time (e.g., within a week)
3. **Architecting** for CD
4. TDD/ATDD/BDD
5. Eliminate flaky (non-deterministic) **tests**
6. Fixing any failing test takes precedence over any other work
7. Test automation
8. Tests are treated as first class citizen: they are developed and reviewed using the same quality standard as code
9. Automate **deployment/release**
10. Aligning testing and production environments
11. Fast and continuous **gathering of user feedback**
12. Optimize **process**
13. Collaborative **culture/organization structure**

Continuous Delivery: Benefits Explained

Lianping Chen

lchen@paddypower.com

@lianpingchen

The presentation represents only my own views and doesn't necessarily reflect those of my employer.