



Appsterdam Guru Sessions

Test Driven Development

July 28, SourceTag offices, Appsterdam

Eloy Duran, @fngtps, Fingertips

Manfred Stienstra, @fngtps, Fingertips

Matt Feigal, @mattfgl, Inter IKEA Systems B.V.

Thanks to:

Apps for Noord Holland, SourceTag,
ICE Amsterdam, Fingertips



“We cannot emphasize strongly enough how liberating it is to work on test-driven code that has thorough test coverage. We find that we can concentrate on the task in hand, confident that we’re doing the right work and that it’s actually quite hard to break the system”

(Freeman and Pryce, [Growing Object Oriented Software, Guided by Tests](#))



[Eloy and Manfred's
presentation]

Zooming out from Test Driven Development

Matt Feigal

@mattfgl, matt.feigal@gmail.com

Inter IKEA Systems B.V.



But we hate testing!

- Boring
- Only gives bad news
- Time consuming
- Expensive
- What are the consequences of this?



A true corporate story (without automated testing)

- Large project. 9 months, 3 teams
- Manual acceptance and integration testing
- Regression errors rise over time, cause trouble
- But project delivers successfully, YAY!
- Now... every change cycle requires 5 people 5 days testing
- Finally paying Indian company to bolt on automated testing of things already built



But those exercises were cool!

Why doesn't everyone do that?

- Businesses don't run on cool
- And It's hard to change habits
- You can already see the value, but you might need to sell the benefits to different audiences:



For Other Developers

- Turns testing into design activity
- Improves your design thinking
 - Don't ask "what code will I write"
 - Ask "How will I know when I've solved the problem?"
- Feedback every few minutes (or less!)
- A clear "finish line"
- *[Input from audience: Also can be a form of documentation]*

For Teams

- Encourages loosely coupled components
- Clear communication about the functionality
- Less likely to break each other's code
- You have something working at all times
- *[Input from audience: it can be done as a game, writing tests to challenge each other's implementation]*

For Project Managers

- Accelerates rather than slows
- Benefit twice from same activity: design and QA *[input from audience: also documentation]*
- Reduce regression errors
- Reduce QA costs at end
- Respond faster to requirements changes
- Prevent developers “gold-plating” features



For Customers

- Built-in quality assurance
- Better long-term maintainability, changeability
- Reduced time (and expense) for changes and bug fixes

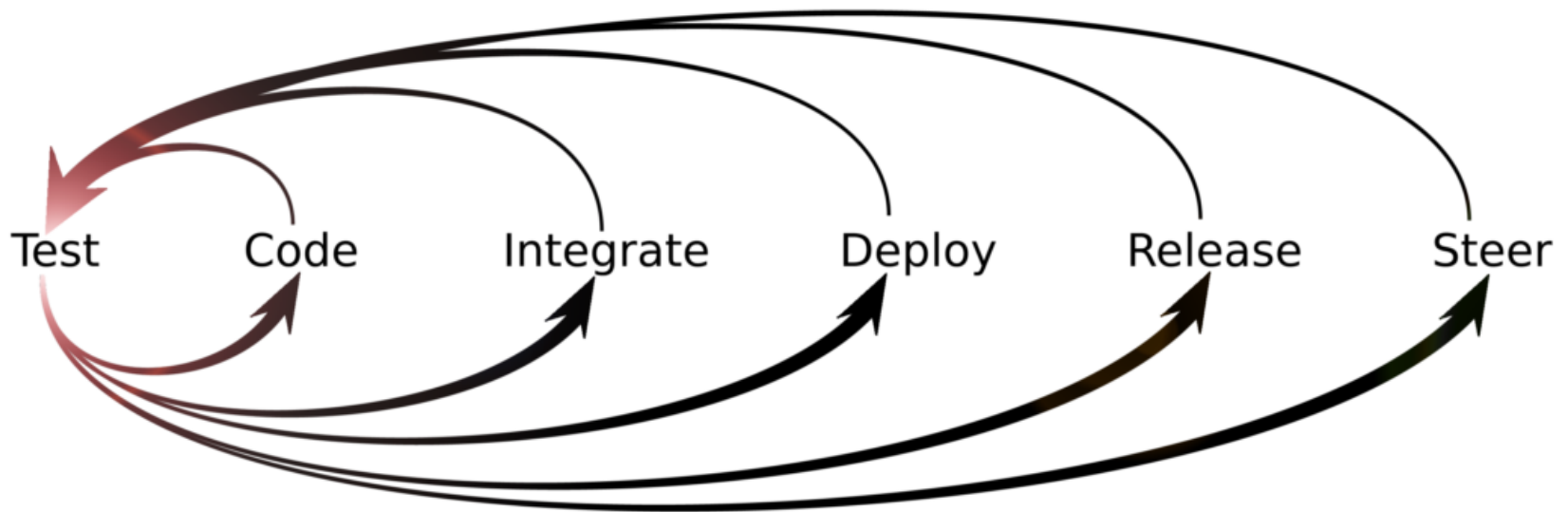
How to get started?

- Simply take responsibility at the team level (it's only a software engineering technique)
- Start small
- Do your research
- Do a few tests to prove the time value
- This idea confuses project managers and customers, approach them **only** with clear evidence of long-term financial value

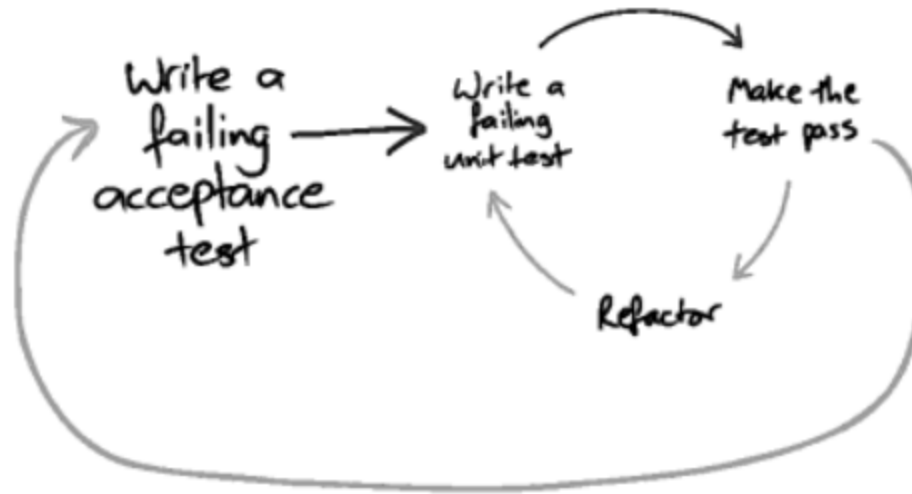
Beyond Unit Tests

(Zooming out again)

- Functional tests, Acceptance tests, end-to-end tests can all be automated
- It's about multiple feedback cycles
- Building a full “skeleton” at beginning pays off



Example of Acceptance test



- First write acceptance tests that exercise functionality you will build
- They will fail 😊
- Underneath that, follow the unit level cycle to develop the feature
- Until Acceptance Tests pass as well.

TDD and Agile projects

- TDD is an excellent tool in XP, Scrum
- Feedback cycles
- “Definition of Done”
- You code just enough to meet the tests
- Gives courage to change.

TDD and non-agile projects

- Simply an excellent engineering practice
- Investment pays off quickly
- Requirements will always change, the regression coverage pays off.
- QA will always be chopped
- The investment in better design, cleaner code always pays off.



TDD and “Brown fields”

- Legacy code = technical debt
- Adding TDD to the areas you need to change can help reduce the risks.
- Check out the “Birthday Greeting Kata”, which includes refactoring existing spaghetti code:
<http://matteo.vaccari.name/blog/archives/154>

References

- String Calculator Kata:
- <http://osherove.com/tdd-kata-1/> (many languages)
-
- Birthday Greeting Kata:
- <http://matteo.vaccari.name/blog/archives/154> (Java) or <http://janxl.blogspot.com/2010/02/birthday-greeting-kata-in-c-net.html> (.NET)
-
- Book: [Test Driven Development](#) , Kent Beck
- Book: [Growing Object Oriented Software, Guided by Tests](#), Freeman and Pryce



PIZZA!

Matt Feigal

@mattfgl, matt.feigal@gmail.com

Inter IKEA Systems B.V.

