

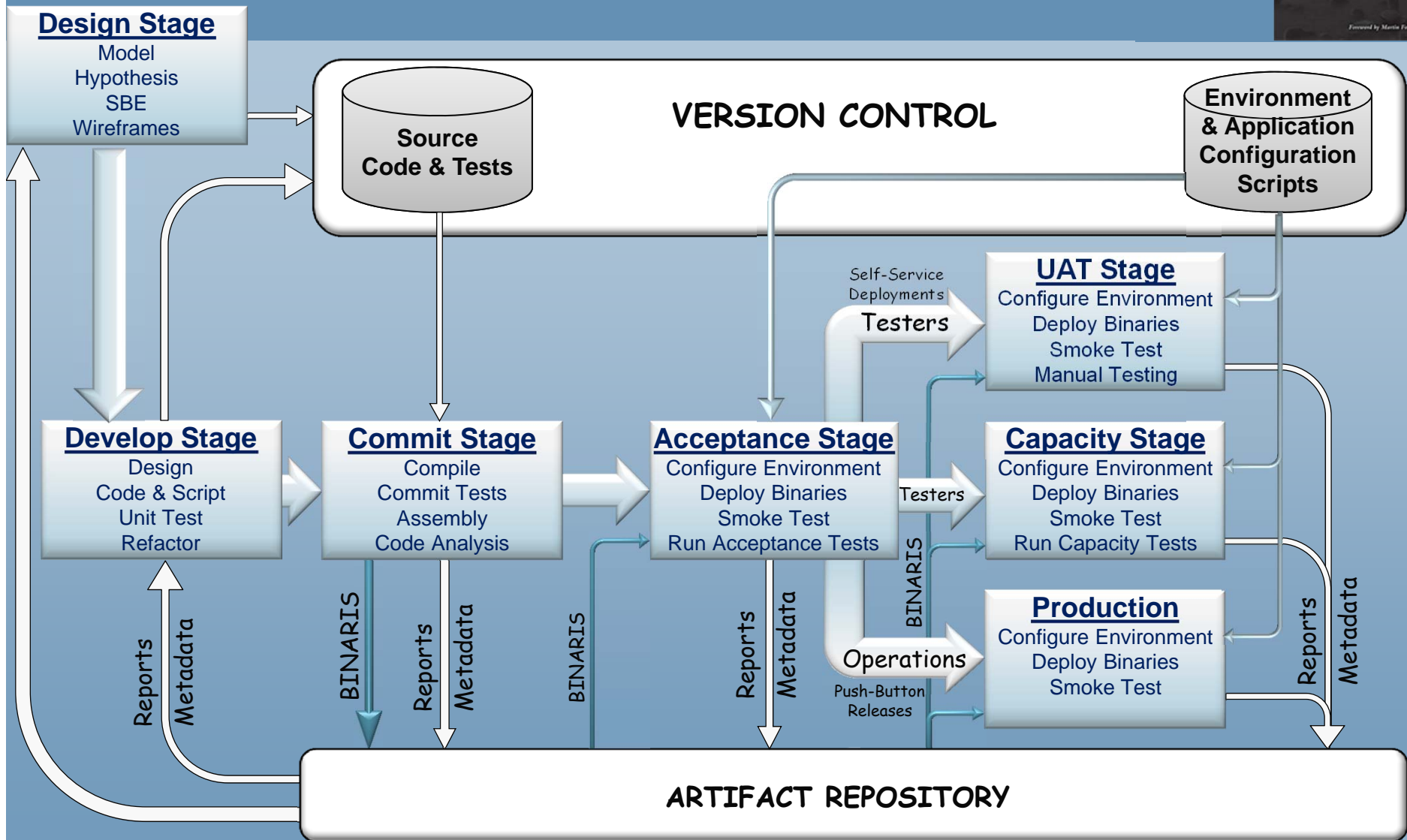
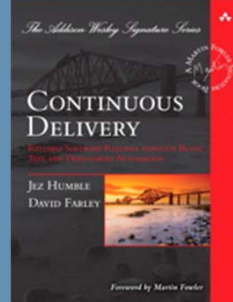
l e a n

software development

Continuous Design

The Other Side of Continuous Delivery

Continuous Delivery



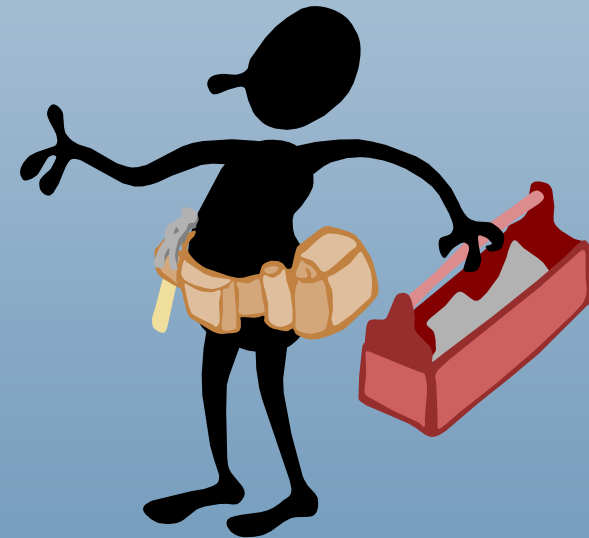
Continuous Delivery Requires Continuous Design



Design Thinking

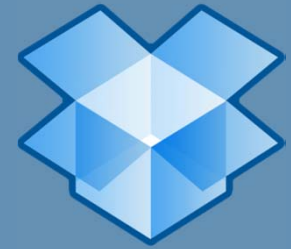


Feedback Tools





*Elegant, simple products
that “Just. Work.”*



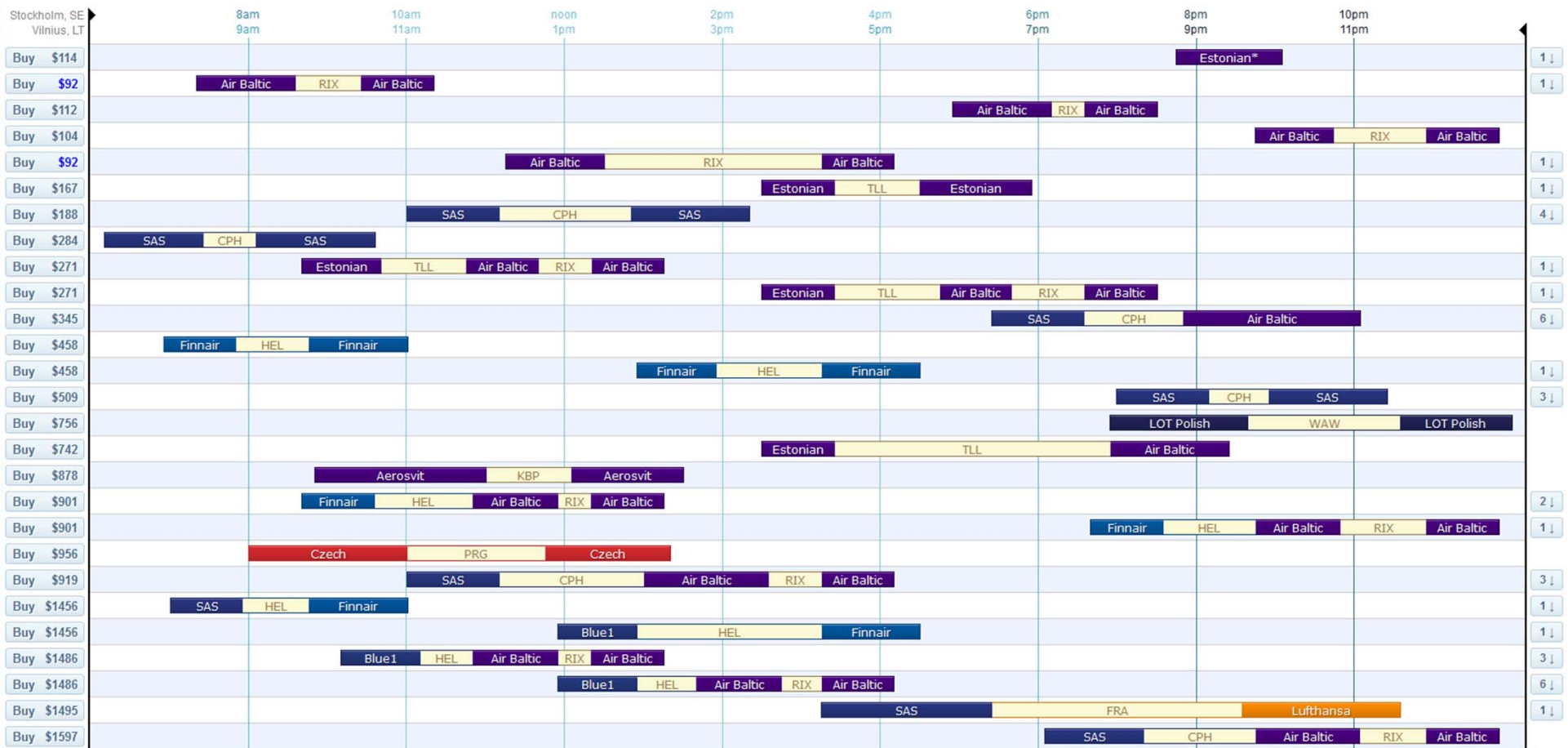
ARN → VNO, Oct 5

+ New Search

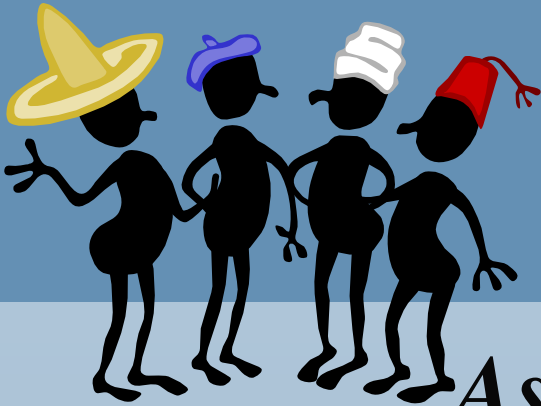
Tweet Like

Sort By: **Agony** Price Stops Departure Time Arrival Time Duration

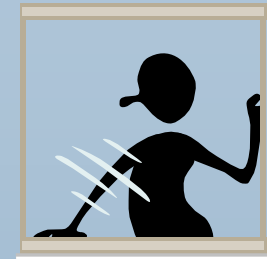
Filter by Time: Show all



Design Thinking



Assemble a Diverse Team



Framing

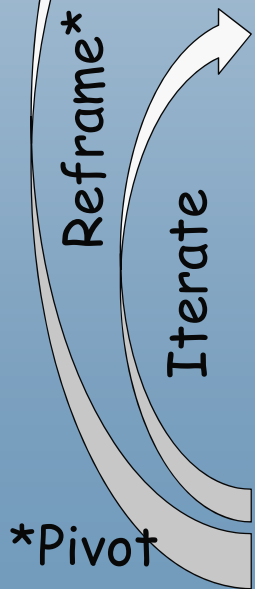
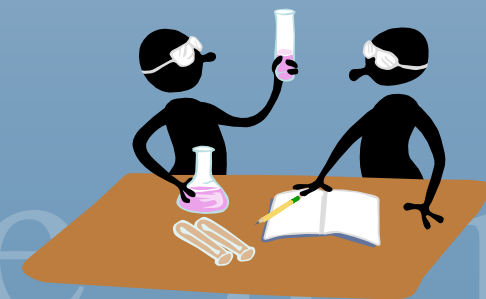
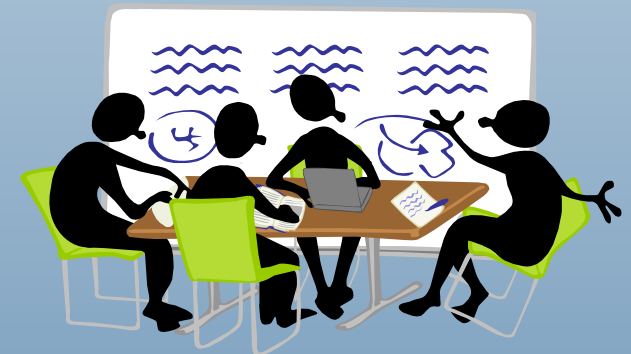
- ✓ Understand the Purpose
- ✓ Observe the Situation
- ✓ Conceptualize the Problem

Ideation

- ✓ Obtain Customer Insights
- ✓ Visualize/Prototype Ideas
- ✓ Develop Hypotheses/Model

Experimentation

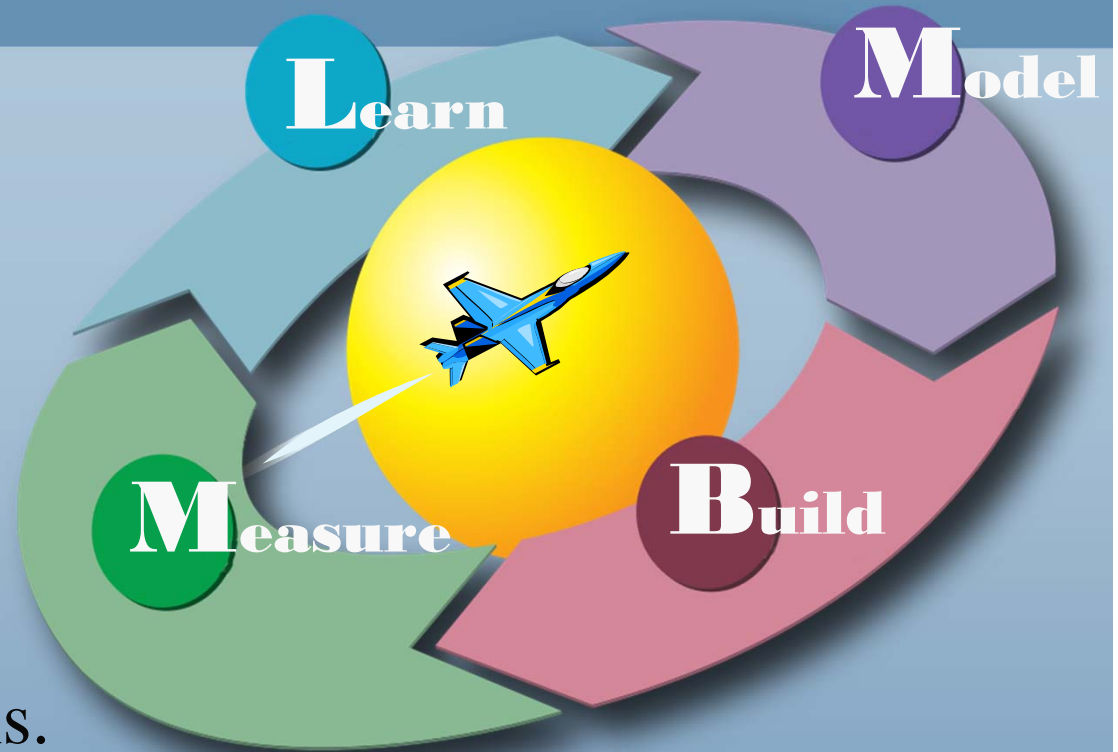
- ✓ Try Tentative Solutions
- ✓ Validate Hypotheses (or not)
- ✓ Redefine the Problem (perhaps)





The Learning Cycle

*Make a little.
Sell a little.
Learn a little.*



The Fastest Learner Wins.

When development programs are managed and executed by a small, experienced, dedicated, decision-empowered team, fast innovators can introduce four times more product in the same amount of time with the same number of people.

*George Stalk,
Competing Against Time*

Organizing for Continuous Design



Good People:

- ✓ Have the capability and experience to do their work well.
- ✓ Have a disciplined approach to work.
- ✓ Don't need to be tightly managed.
- ✓ Fulfill their commitments.



Product Metrics:

- ✓ Success is measured in terms customer adoption/satisfaction.

Whole Team:

- ✓ Team members are dedicated to and stay with a system
- ✓ Teams include *all functions*: marketing, design, development, testing, operations, support, etc.
- ✓ Teams members are competent in their functional areas.
- ✓ Teams make scheduling and content decisions internally.



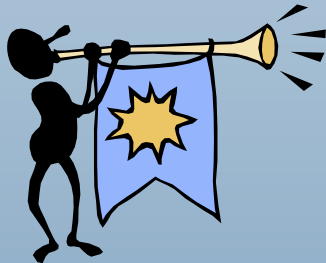
Start with Customers



amazon.com[®]

Working Backward

1. Write a Press Release



2. Write a list of FAQ's (and answers)

3. Describe the Customer Experience

4. Write a User Manual.

http://www.allthingsdistributed.com/2006/11/working_backwards.html

PRESS RELEASE

Heading –

Name the product in a way the reader (i.e. your target customers) will understand.

Sub-Heading –

Describe who the market for the product is and what benefit they get. One sentence only underneath the title.

Summary –

Give a summary of the product and the benefit. Assume the reader will not read anything else so make this good.

Problem -

Describe the problem your product solves.

Solution –

Describe how your product elegantly solves the problem.

Quote from You –

A quote from a spokesperson in your company.

How to Get Started –

Describe how easy it is to get started.

Customer Quote –

Provide a quote from a hypothetical customer that describes how they experienced the benefit.

Closing and Call to Action –

Wrap it up and give pointers where the reader should go next.

Disruptive Design



GE Healthcare



The Vscan: \$8000 Ultrasound unit the size of a mobile phone. Based on designs originating in China, it is revolutionizing global healthcare.

“We realized that the biggest impediment was that we were selling what we were making [rather than] making what the customers here needed.”*

*V. Raja, president and CEO of GE Healthcare-South Asia.

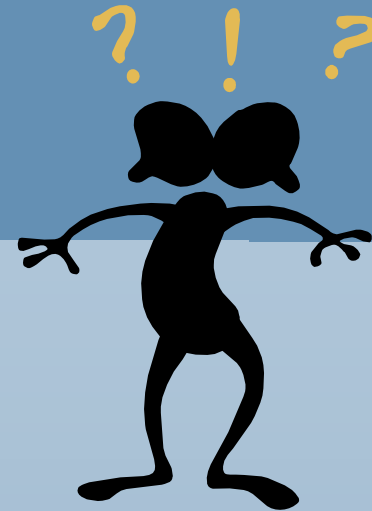
“Our engineering and marketing teams now interact closely with the customers here [in India] to understand their requirements. We look at their work flow, their environmental limitations, their profitability issues and other factors and we then price, design and manufacture the products accordingly”**

**Ashish Shah, General Manager, GE Healthcare Technology - India



The MAC-i: ~\$500 - EKG's for Rs 9

Which One?



“It’s not the customers’ job to know what they want.”



It’s hard to refactor:

- ✓ *a book*
- ✓ *hardware*
- ✓ *first impressions*



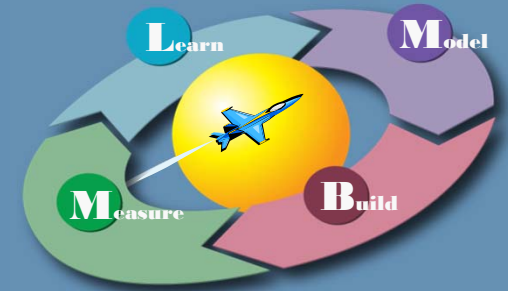
Don’t Guess: Measure

Google

How much can you afford to lose?

l e a n

Minimum Viable Product



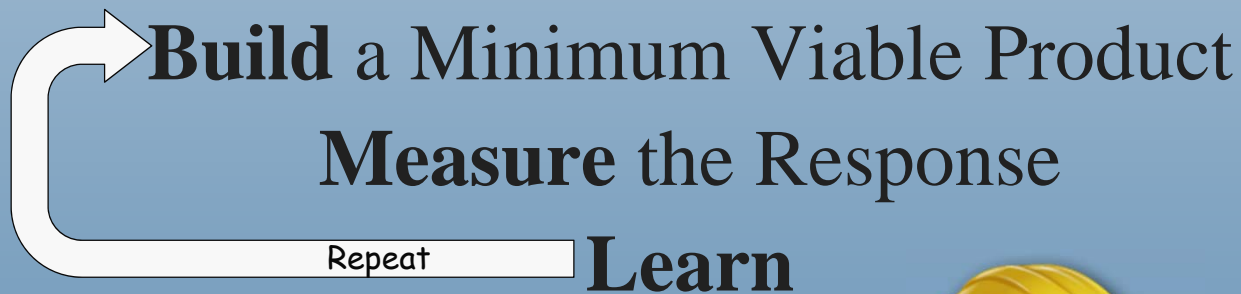
What is the biggest waste in Software Development?

Building the Wrong Thing

What's the 2nd biggest waste in Software Development?

Complexity caused by all that extra stuff.

How do we stop building the wrong thing?



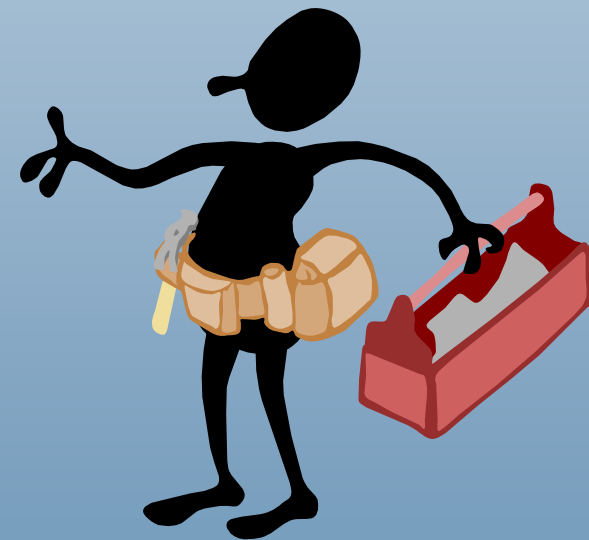
Continuous Delivery Requires Continuous Design



Design Thinking



Feedback Tools



Actionable Metrics



What metrics are most useful for product development?

- ✓ Metrics which demonstrate clear *cause and effect*.

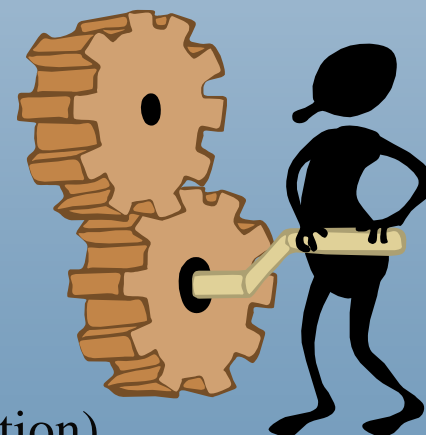
Avoid “Vanity” Metrics

- ✓ Eg: Unique visits, page views, time on site, downloads, even revenue.
- ✓ These metrics are not actionable - they say nothing of *cause and effect*.

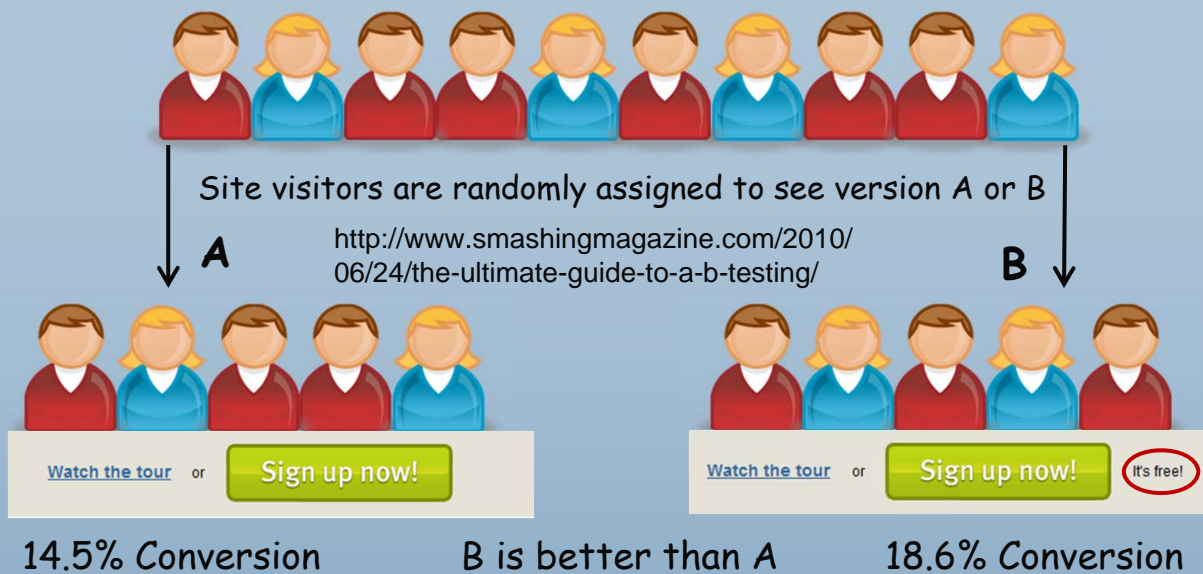
Use “Innovation Accounting” to Drive Product Development

1. Start with a Hypothesis – A Business Growth Model
 - ✓ Sign-up and trial rates
 - ✓ Conversion rates
 - ✓ Customer lifetime value

(These metrics will differ based on the business.)
2. Establish a baseline – with a Minimum Viable Product
3. Target every initiative at improving a growth metric
 - ✓ Focus on the most important growth drivers
4. Do not add features without a split test (or equivalent validation)
 - ✓ Definition of done: The *value* of the feature has been validated
 - ✓ The whole team is engaged in validating the value of each feature.



A/B Experiments



Amazon.com Case Study

Amazon's Greg Linden created a prototype which gave personalized recommendations based on items in the shopping cart. The feature was opposed by a marketing SVP who told Greg to stop, but Greg created a test anyway. He was allowed to push it live. The feature 'won' by such a wide margin that it was immediately adopted, increasing sales by an estimated 3%. <http://glinden.blogspot.com/2006/04/early-amazon-shopping-cart.html>

Don't Miss this excellent paper:

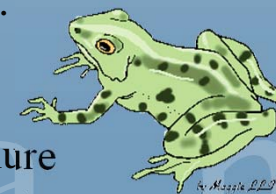
Online Experimentation at Microsoft
by Kohavi, Crook, & Longbotham

Presented at KDD 2009

(Knowledge Discovery & Data Mining)

<http://exp-platform.com/expMicrosoft.aspx>

1. Most of the time the guess about how customers will behave will be wrong – even for experts!
2. Test early – don't waste a lot of time on in-depth analysis or planning the perfect design.
3. Test often – most experiments don't tell much. [You have kiss a lot of frogs to find a prince.]
4. A failed experiment is not a failure – it's a learning opportunity. The only failures are failure to learn or failing to conduct a good experiment.



Cohort Metrics



A cohort is a group of people who have a similar experience.

- ✓ Cohort 1: Customers who look up account balances on-line
- ✓ Cohort 2: Customers who ask for account balances by phone

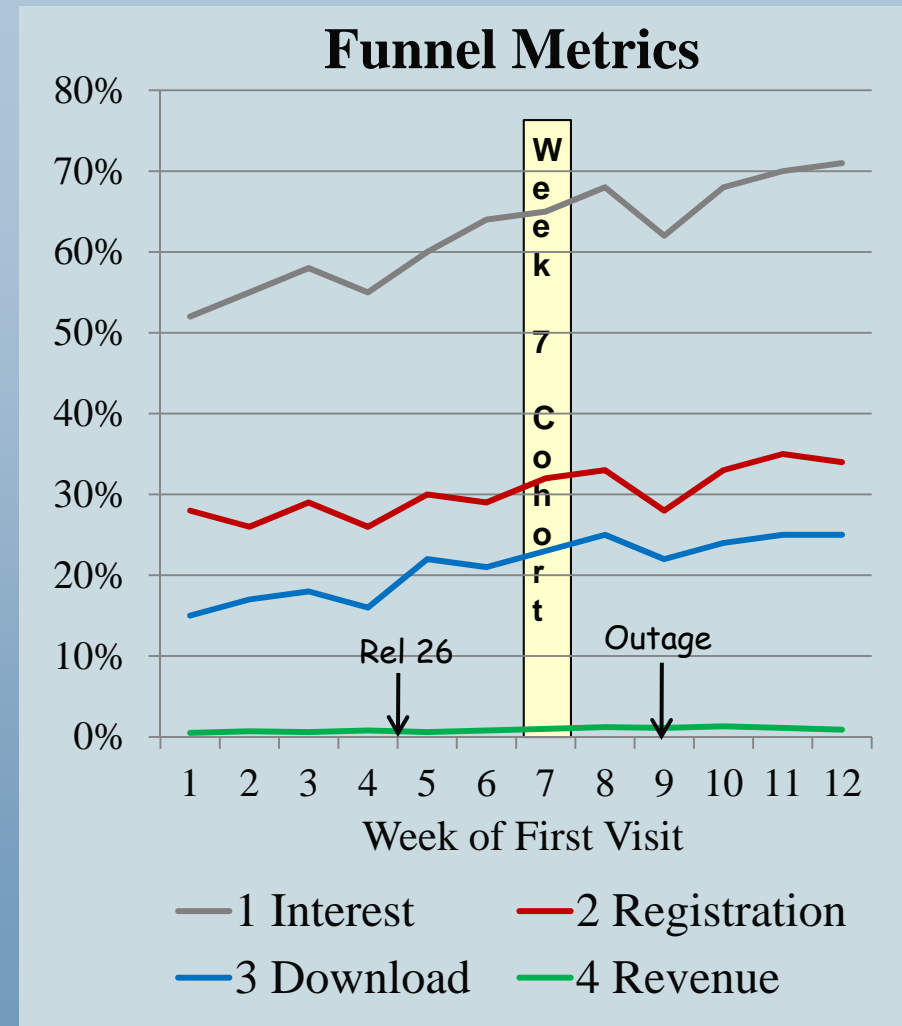
Track customer experience by cohort => actionable metrics.

- ✓ Cohort 1: Satisfied with the service
- ✓ Cohort 2: Unsatisfied with the service
 - ✗ Improve phone retrieval of balances

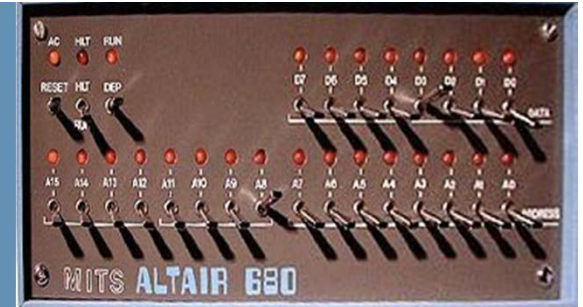
What cohort metrics are most useful for product development?

1. Create a cohort of new users every week (or month).
2. Track desired behavior of each cohort against product events and changes

Eric Ries, *Startup Lessons Learned*



Feature Toggles

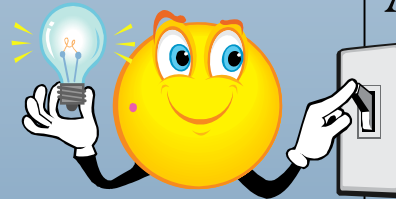


Why Would You Want to Toggle a Feature?

- ✓ Avoid feature branches and associated merge problems.
- ✓ Enable continuous delivery of partially done features.
- ✓ Allow any feature to be switched on and off “on demand.”

Especially Useful For:

- ✓ A/B Experiments
- ✓ Canary Release (Rollback)
- ✓ Branch by Abstraction
- ✓ Levers to Reduce Traffic
- ✓ Etc.



Also known as:

Feature Switches
Feature Bits
Feature Flags
Feature Flippers

How do Feature Toggles Work?

1. Wrap the entrance to the feature with toggle code.
2. Store the toggle's value (on/off) in a configuration file.
3. At runtime, the toggle executes the feature only if its value is 'on'.

See: <http://blog.disqus.com/post/789540337/partial-deployment-with-feature-switches>

Canary Releasing

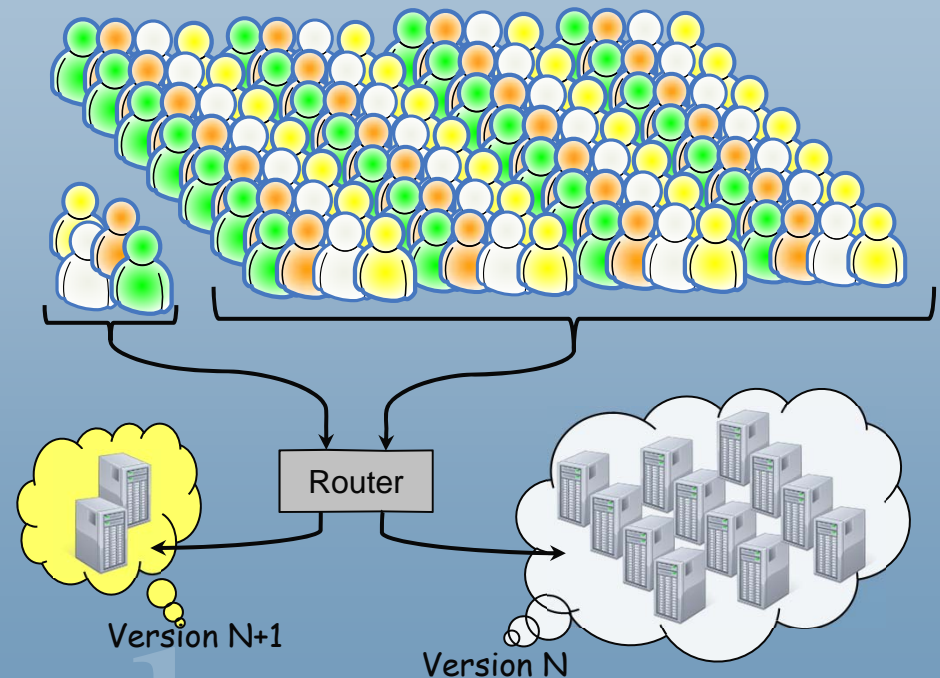


Google Chrome Canary

Years ago, miners brought canaries into mines because the birds react quickly to methane and carbon monoxide gas. If the canaries got sick or stopped singing, miners were alerted to danger.

Canary Release with Immune System:

1. Automate deployment rollout
 - ✓ Script deploys new version to a few users
 - ✓ Gradually add users in a controlled manner
2. Be able to tell a good change from a bad change quickly
 - ✓ Immune System monitors a few statistically significant metrics and alerts if out of range
3. Roll back a bad change quickly
 - ✓ Immune System automatically rolls back deployment if a critical threshold is crossed
4. Don't have the same problem twice
 - ✓ Immediate root cause analysis upon rollback



<http://engineering.imvu.com/2010/04/09/imvus-approach-to-integrating-quality-assurance-with-continuous-deployment/>
and <http://continuousdelivery.com/patterns>

The Five Why's



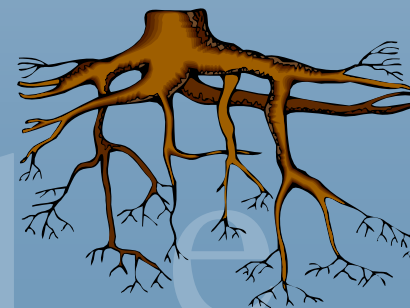
Web site goes down

1. Why was the website down?
 - ✓ The CPU utilization on all our front-end servers went to 100%
2. Why did the CPU usage spike?
 - ✓ A new bit of code contained an infinite loop!
3. Why did that code get written?
 - ✓ So-and-so made a mistake
4. Why did his mistake get checked in?
 - ✓ He didn't write a unit test for the feature
5. Why didn't he write a unit test?
 - ✓ He's a new employee, and he was not properly trained in TDD

Countermeasures

At every level of the analysis:

1. Bring the site back up
2. Remove the bad code
3. Help so-and-so understand why his code doesn't work as written
4. Train so-and-so in the principles of TDD
5. Change the new engineer orientation to include TDD



From Eric Ries:

<http://www.startuplessonslearned.com/2008/11/five-whys.html>

Lessons from Innovative Companies



It's not about Money.

*Solve the customer's problem
and the revenue will show up.*



Focus.

It's best to do one thing really, really well.



Think Long Term.

Be willing to be misunderstood.



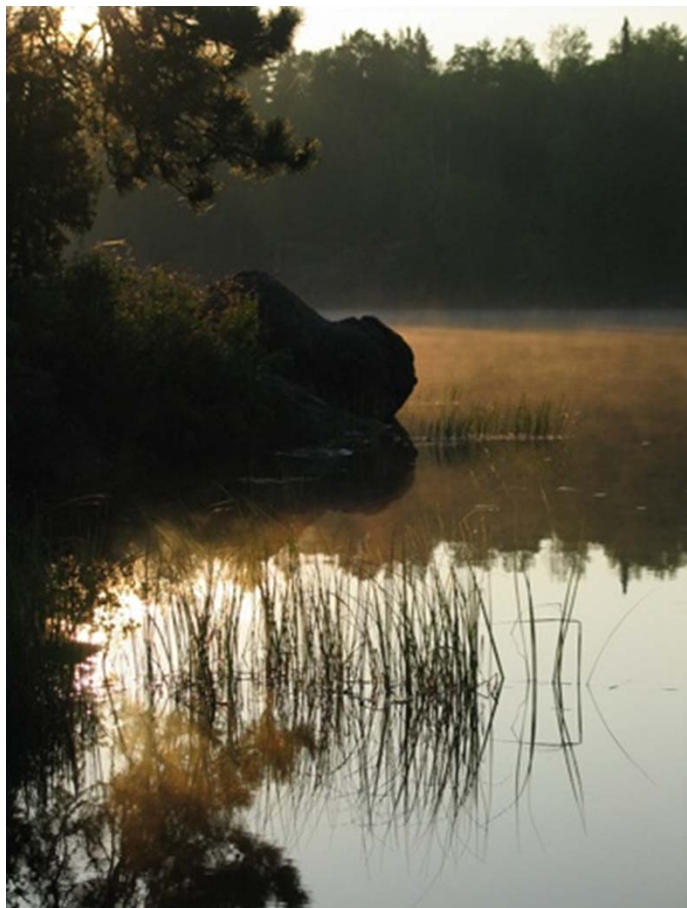
Build Things that 'Just. Work.'

Elegant, simple products that fit peoples lives.



Hire Good People & Leave Them Alone.

*Good people want to do their jobs in their own
way. Mistakes will be made. But these are not
as serious as the mistakes management will
make if it tells people how to do their jobs.*



l e a n

software development

Thank You!

More Information: www.poppendieck.com