

Agile2010

Nashville

Gaylord Opryland Resort

August 9 - 13, 2010



Feature Injection and Test Driven Requirements

Amr Elssamadisy

When Chris Matts asked me to write this article I jumped at the opportunity. The concepts presented in this comic book describe several powerful thinking tools to take software development to the next level and I predict that they will be part of every Agilist's vocabulary in a few years. In this article I'll share one of the insights I've gained from these ideas.

Test Driven Requirements

In the Agile community, test driven requirements – the practice of writing requirements as examples and writing tests for these examples (also known story driven development and closely related to behavior driven development) – is one of the least implemented practices. It is also, in my personal experience, one of the most powerful practices to improve development quality, speed, and flexibility. Simply put, test driven requirements takes the same idea in test driven development – writing tests before you write the code – one level higher. Test driven requirements entails starting each iteration with a set of requirements written as tests that exemplify how the system will function after the features to be built in this iteration are developed.

Test driven requirements is probably the least implemented of practices because of its difficulty and long cycle time for benefits to appear. It requires analysts - who are largely left out of the majority of Agile literature - testers, and developers to work together, cross-pollinate new skills, and share responsibilities. It also requires the same discipline needed for test driven development and frequently requires refactoring of the current system so that it becomes 'testable'. Experts in the field have written about this practice and its benefits but the reasoning to adopt this skill is largely anecdotal.

Feature Injection and Software as an Information System

Feature injection, as you will read in this book, depends on an viewing software development as an information system instead of a manufacturing system. By seeing software development as an information system the utility of different practices take on new light. The effectiveness of a team increases with the amount of information that is understood and used. The more information you have, the better decisions you can make, so investments that produce information become much more compelling because they enable your team to make better decisions.

One of the profound insights from viewing software in this manner is that information flows in the opposite direction of work product. That is, if software is created by performing analysis, programming, testing, and deploying, then information flows in the opposite direction – from deploying to testing to programming to analysis. For example, when a defect is found in testing, this is a piece of new information that is communicated to development and possibly analysis depending on the type of defect. Why is this a profound insight? Well, if we want to maximize information, then we should perform all of the later development steps such as testing and deployment as soon as possible to make the information available.

Software development as an information system implies maximizing our use of information in the system. When analysts do their work in a traditional development process, examples are created and discarded during the analysis process. They are recreated AFTER development is done by testers and only at that point in time is the information introduced into the system as defects with reproduction steps. This throws a wrench into planning because now you have a random event creating work for developers.

This information was *in* the system when analysts first worked out their models and discarded. Instead of throwing it away, feature injection suggests that these examples be kept and communicated to the development team. The information is not lost. Test driven requirements does exactly this by codifying the requirements as executable examples.

Putting It All Together – A Compelling Reason to Use Feature Injection

Feature injection largely overlaps with test driven requirements in practice, but is based on a different model and different reasoning which is much more compelling than test driven requirements. Here's how feature injection helps explain why teams should spend the effort to learn, adopt, and adapt test driven requirements:

1. Feature injection gives a different reason than “it works great for me, try it for yourself” to writing requirements as examples and codifying them as tests. Writing your requirements as executable examples (tests) passes this information from analysis to development instead of waiting for it to be (re)discovered in the testing phase.
2. Feature injection takes this even further by making example-building the primary mechanism for building your model. Models are refined by introducing more and more examples. Again, this puts increased importance and meaning to test driven requirements since as the model ages and teams members leave, the tests act as an executable model of the world.
3. Finally, feature injection, when taken with the other tools from the real options toolbox, form a more cohesive model for why many practices are valuable. Understanding *why* is the key to moving forward from adopting a practices to adapting practices for specific environments.

These comics you have in your hands are simple to read, and yet describe some very powerful concepts. Take the chance to read them carefully and use them liberally.

Good luck,

Amr Elssamadisy

Introduction

Real Options are derived from Financial Mathematics and Applied Psychology. As we travel from the specialist financial markets to the real world, many of the assumptions (or axioms) in Financial Maths fall apart and we are left with our “rules”...

Options have value

Options expire

Never commit early unless you know why.

Small enough to fit on the back of a business card, big enough to change the world. ;-)

The real option based 2020 Vision Conference series kicks off next year. You can register interest and be keep up to speed with news by joining the “2020 Vision” groups on Facebook or LinkedIn.

You can join in the discussion at

http://finance.groups.yahoo.com/group/real_options_discussion

Creative Team

Words & Pictures – Chris Matts

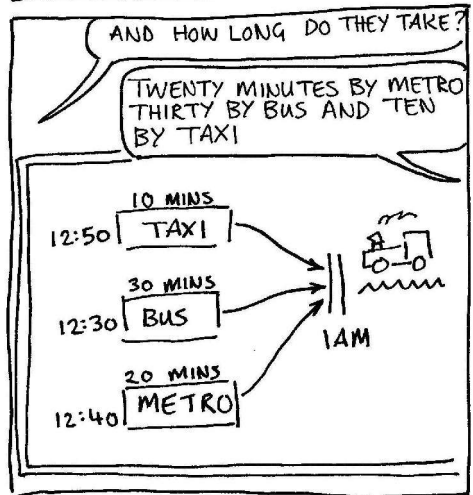
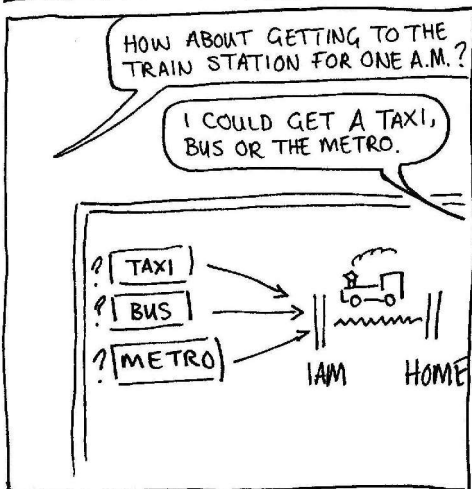
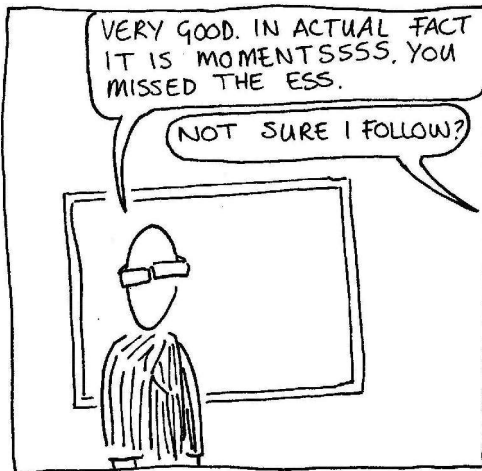
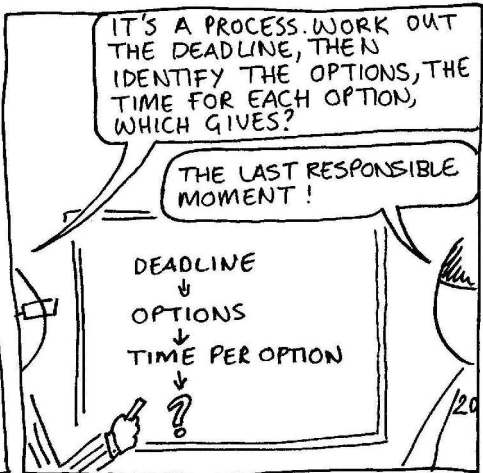
Cover Artist – Andy Belanger

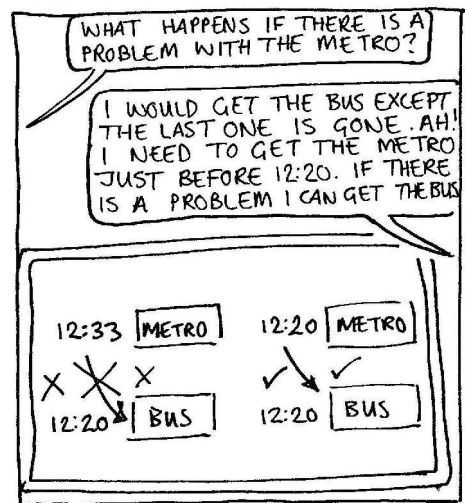
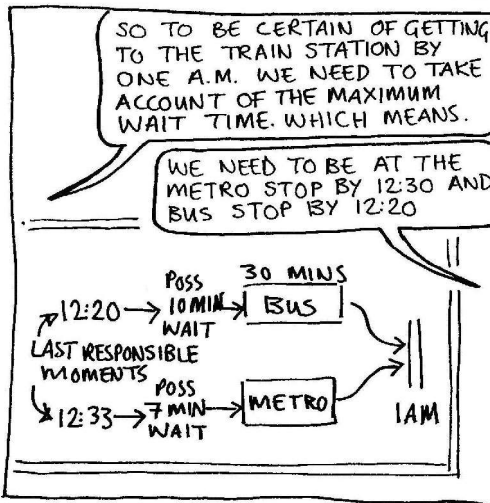
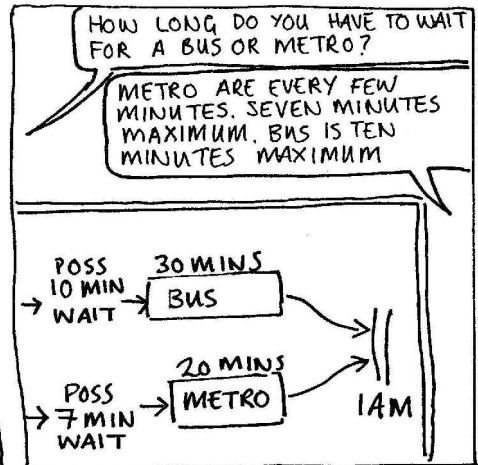
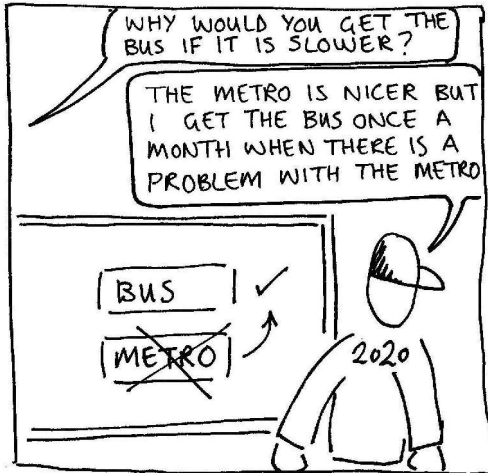
(www.andybelanger.com)

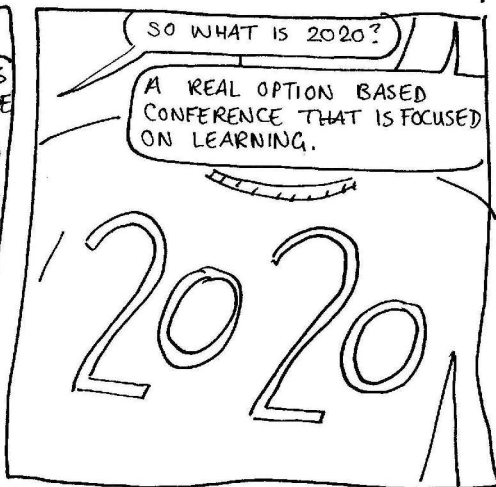
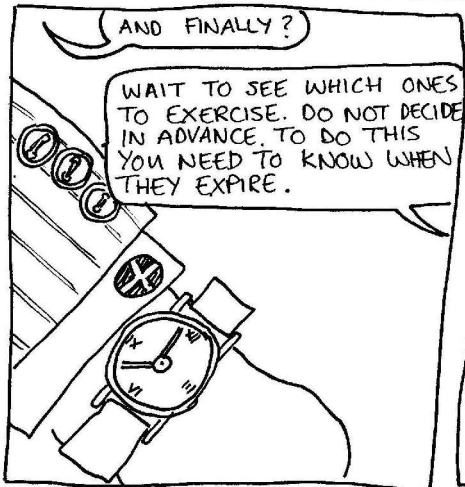
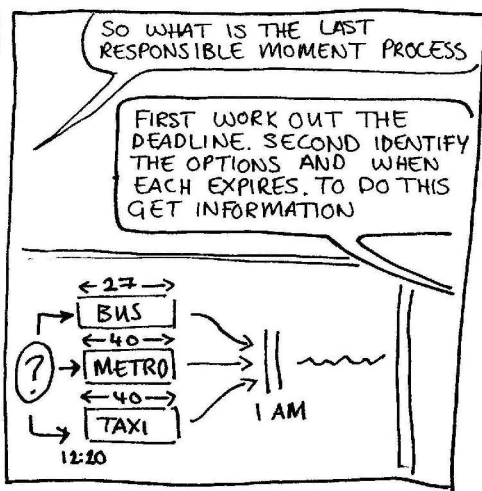
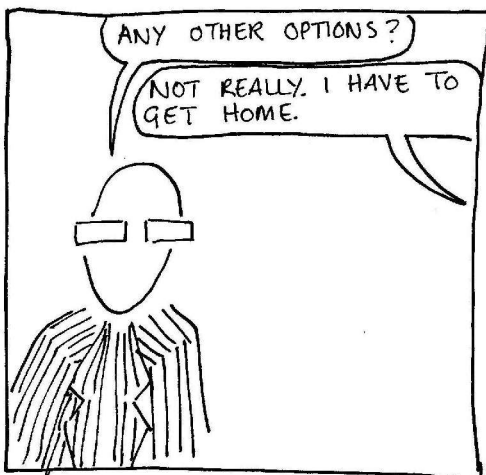
Cover Design – James Schiavi

Editor – Amr Elssamadisy

Translator – Olav Maasen







2009 UK LEAN & KANBAN CONFERENCE

Join us at the prestigious RSA in London for the first annual

UK Lean and Kanban software development conference

27th - 29th September, 2009

The conference, which will be held over two and a half days from 27 through to 29 September brings together the leading innovators, practitioners, and influential thought leaders in Lean software management.

The Lean and Kanban 2009 event was created to drive further progress in software development by providing a unique opportunity to share best practices and innovation within the growing community of software engineers, managers, and executives adopting Lean and Kanban systems. It provides a unique opportunity to meet with and learn directly from industry luminaries and practitioners from around the world.

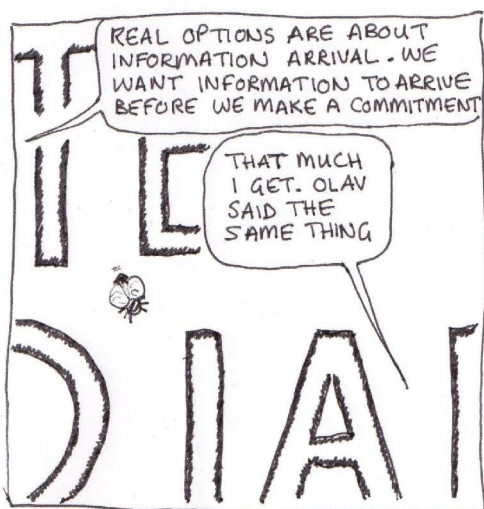
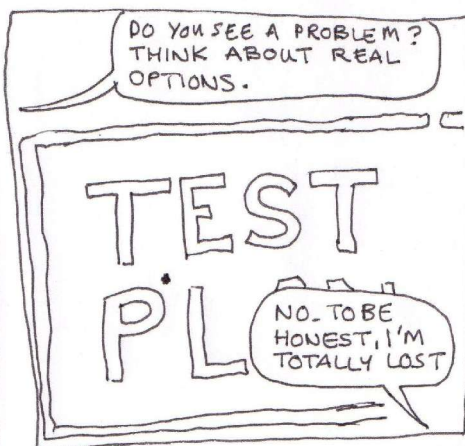
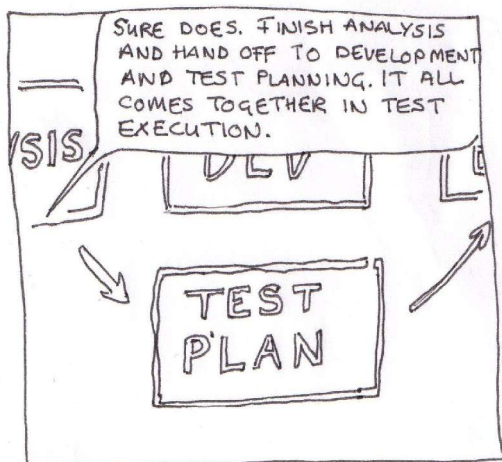
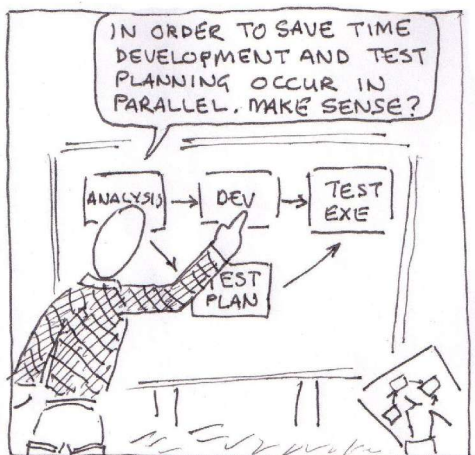
Speakers include:

- Mary Poppendieck
- Don Reinertsen
- David J Anderson
- John Seddon
- Jeff Patton
- Corey Ladas
- Kenji Hiranabe
- Hal Macomber
- Alan Shalloway

Further details at
www.ukleanconference.com

Or contact Jason Smith at
conference@indigoblue.co.uk or
020 7692 4832





RIGHT, NOW WHERE CAN INFORMATION ARRIVE IN THE PROCESS?

ANALYSIS, DEV, TEST PLAN, TEST EXECUTION, EVERYWHERE. THOUGH LESS SO IN TEST EXECUTION

AND ALL THIS INFORMATION NEEDS TO FLOW THROUGH DEVELOPMENT. LET'S LOOK AT INFORMATION ARRIVING IN ANALYSIS. I'LL LABEL IT "I_A"

LOOKS OK.

NOW THE INFORMATION ARRIVING INTO DEVELOPMENT. I'LL LABEL THAT "I_D"

THAT LOOKS FINE AS WELL

NOW LOOK AT INFORMATION ARRIVING INTO THE TEST PLANNING PHASE. I'LL CALL IT "I_T".

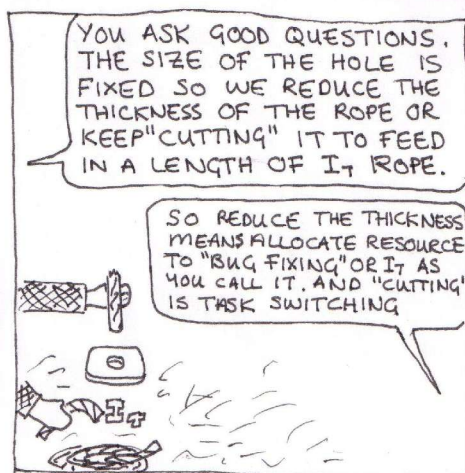
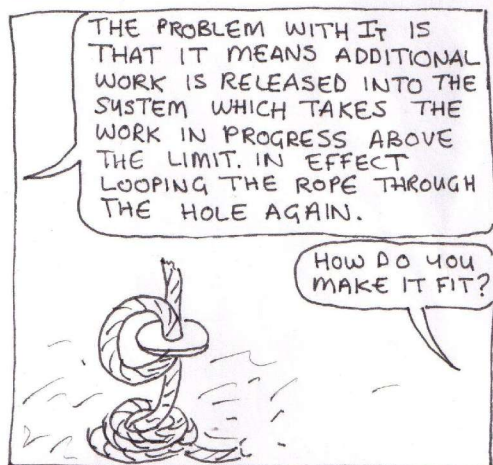
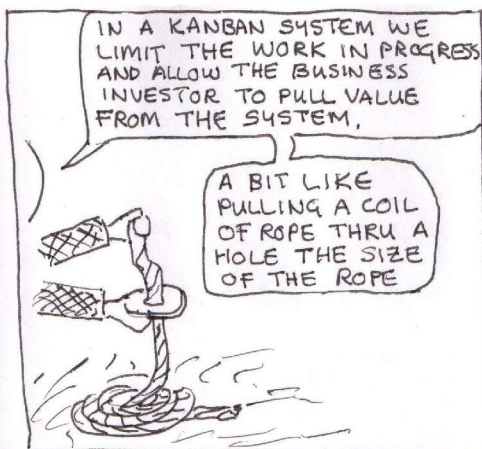
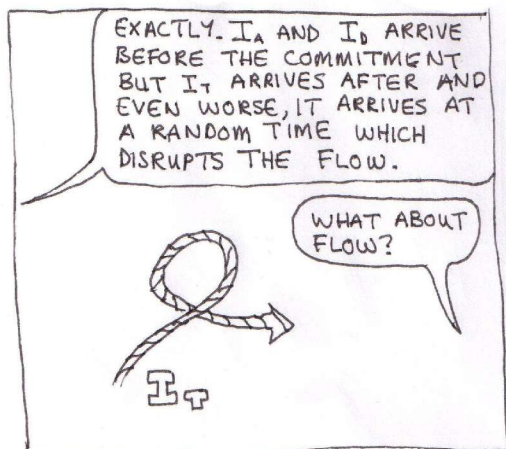
OOOH! HOW DOES IT GET INTO DEV?

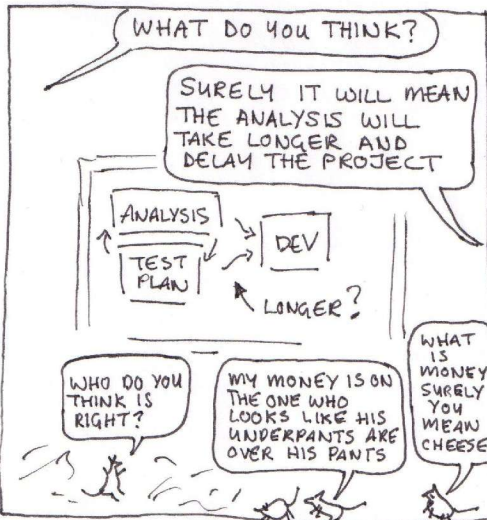
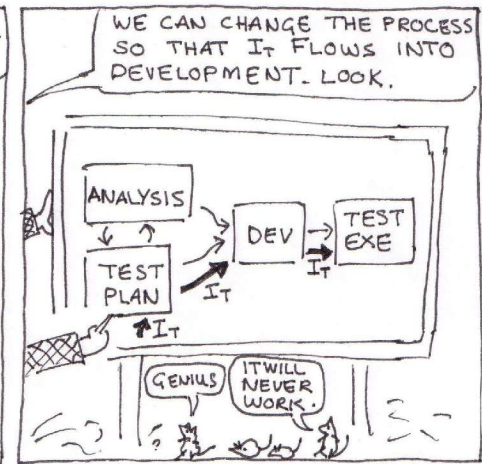
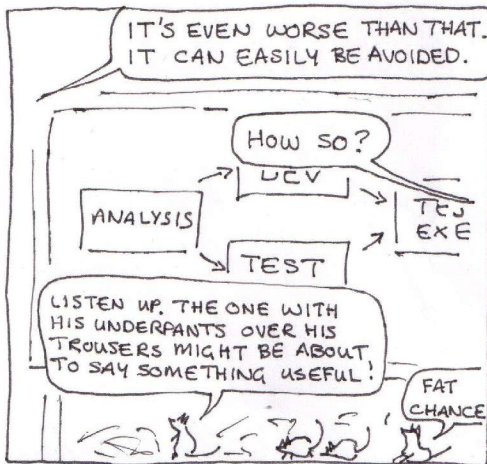
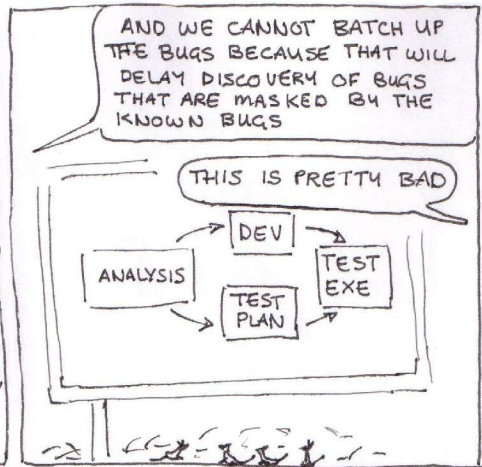
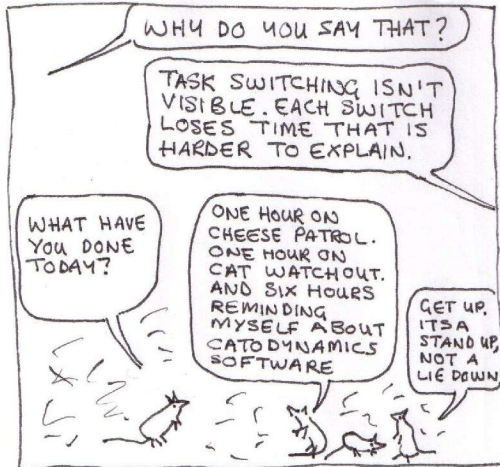
THE INFORMATION WILL GET INTO DEVELOPMENT BUT IT IS AS BUGS THAT DISRUPT THE FLOW.

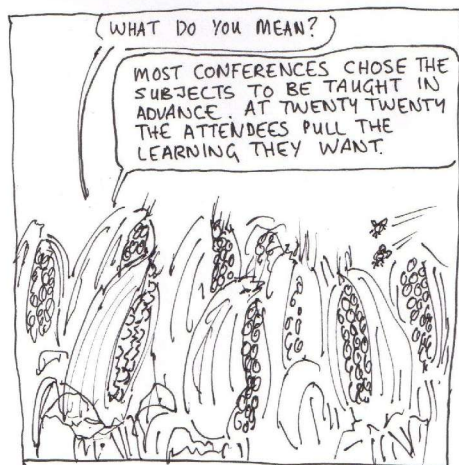
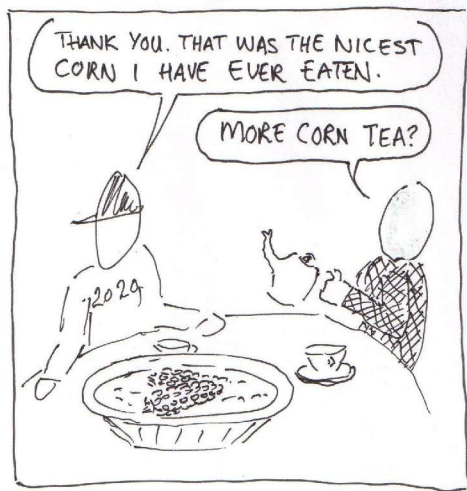
WHAT DO YOU MEAN BY DISRUPT THE FLOW

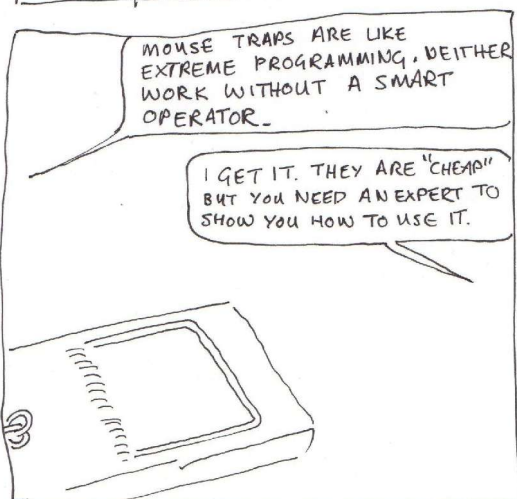
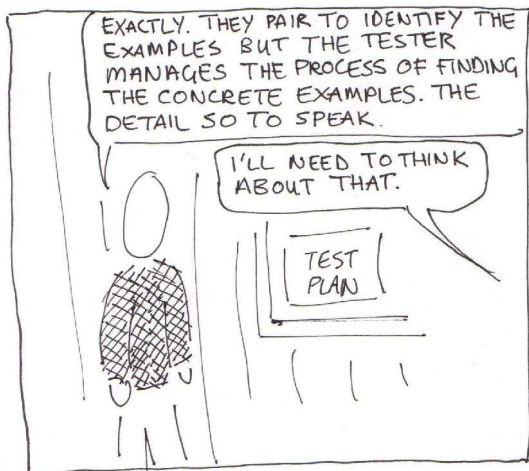
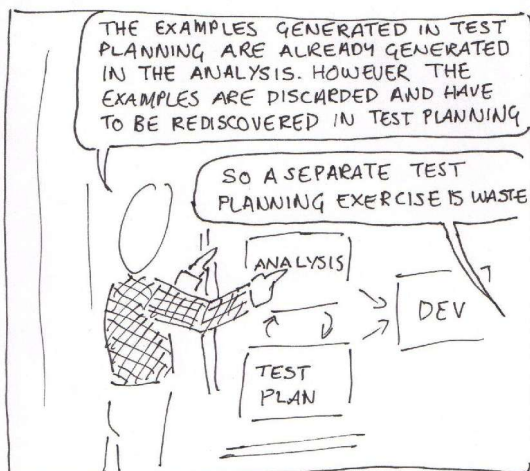
WELL I_A, I_D AND I_T NEED TO BE INCORPORATED INTO THE DEVELOPMENT. THIS CAN HAPPEN IN AN ORDERLY MANNER OR IN A RANDOM UNCONTROLLED MANNER.

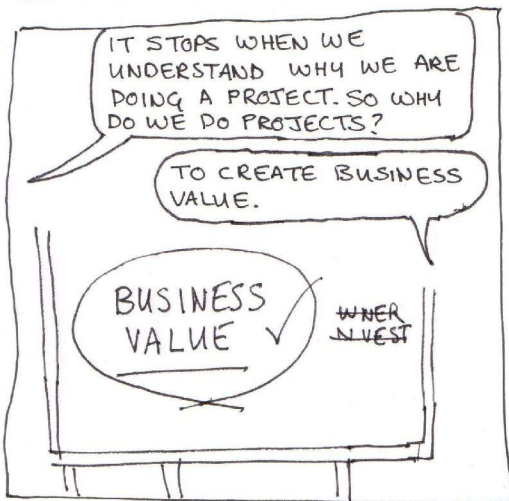
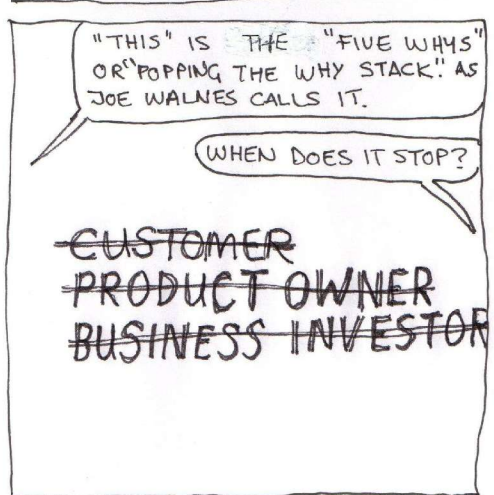
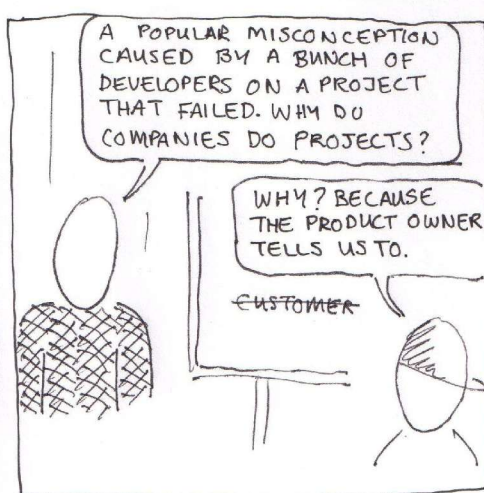
SO I_T ARRIVES RANDOMLY INTO DEVELOPMENT, WHEREAS I_A AND I_D ARRIVE BEFORE ANY COMMITMENTS.

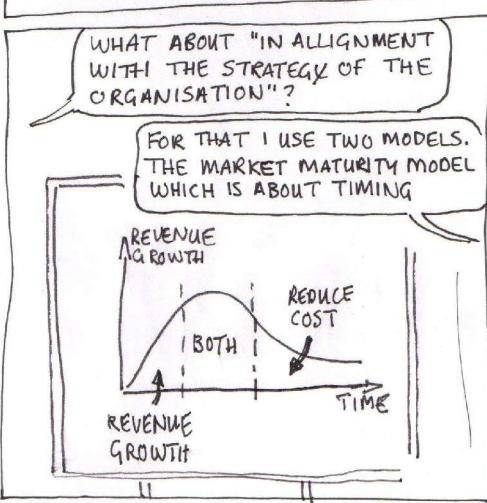
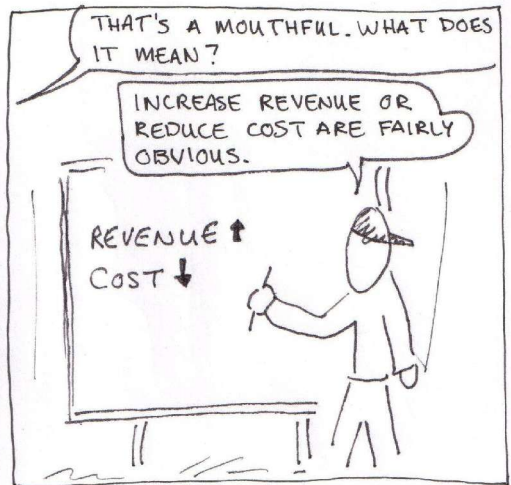








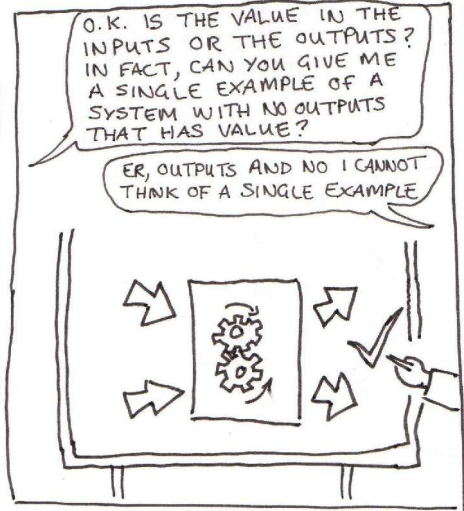
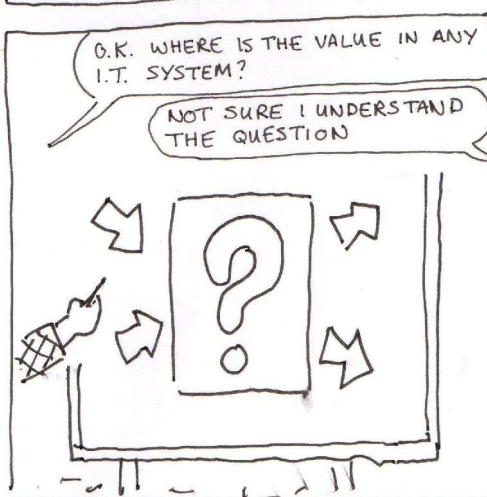
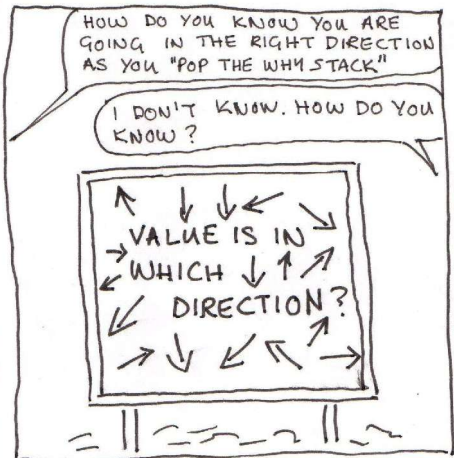
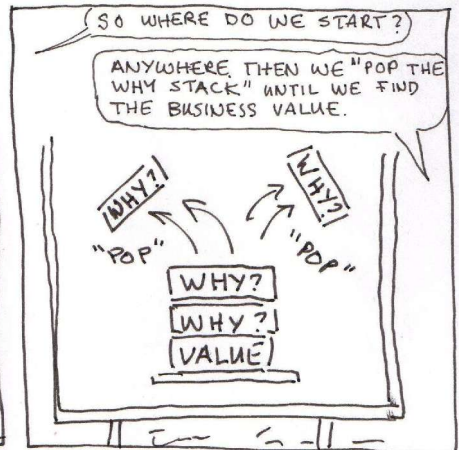


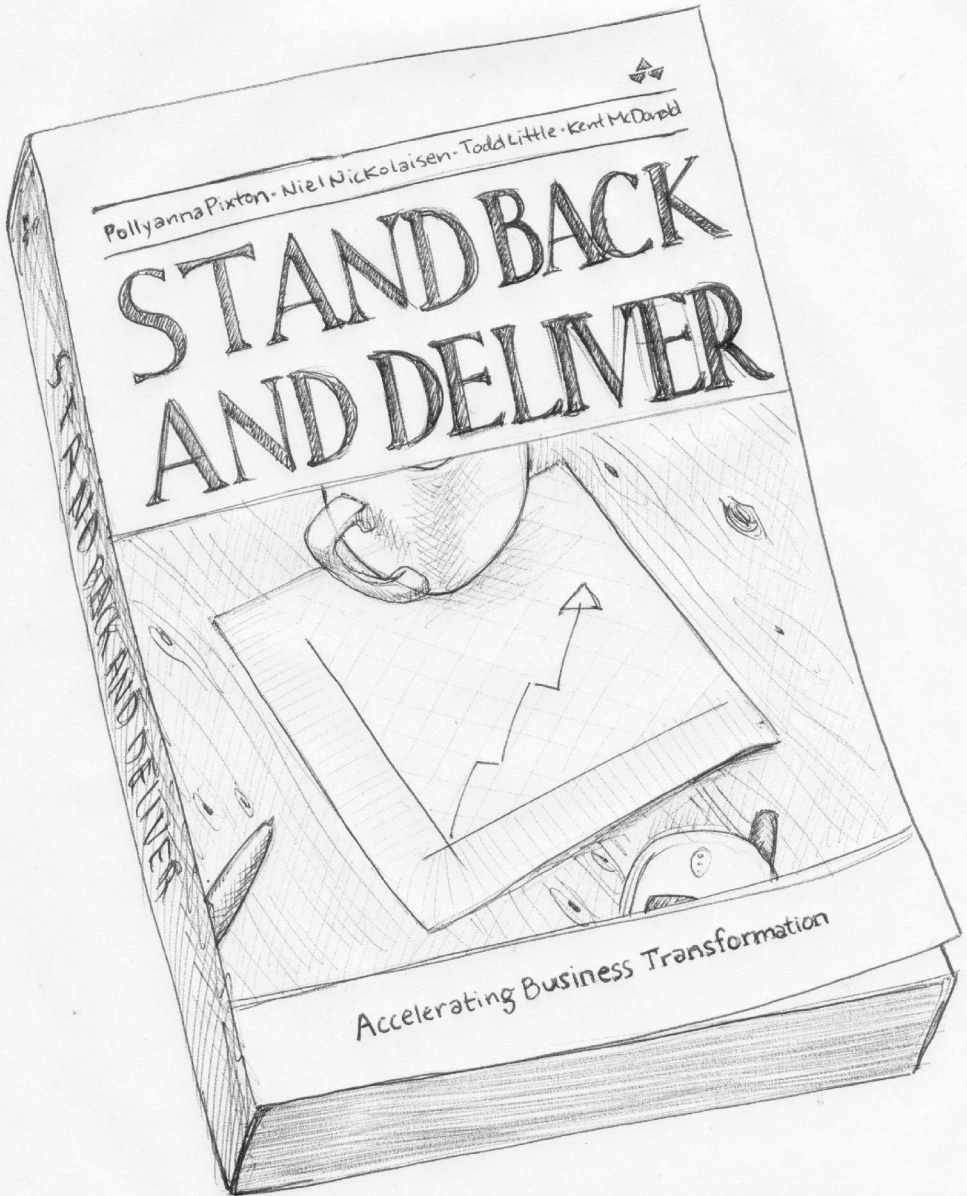


AND THE SECOND MODEL?

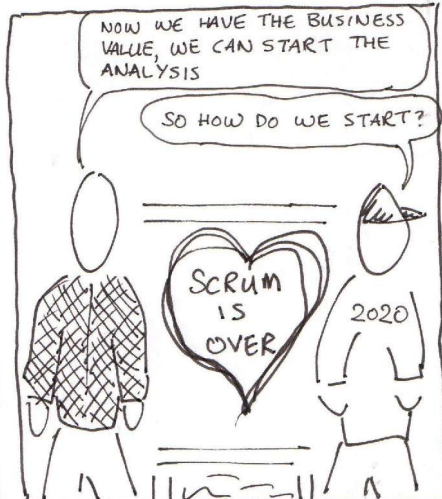
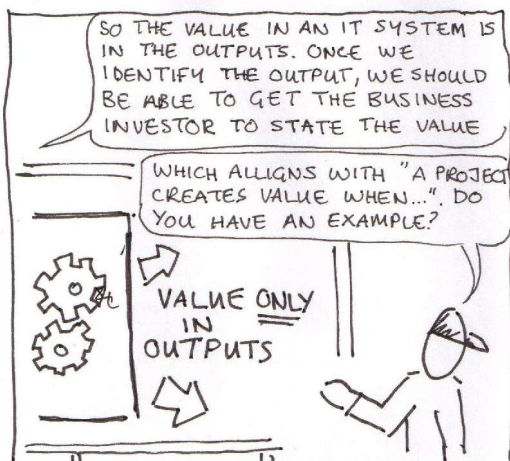
THE MIEL NICKELISEN MODEL. TO DETERMINE HOW TO INVEST

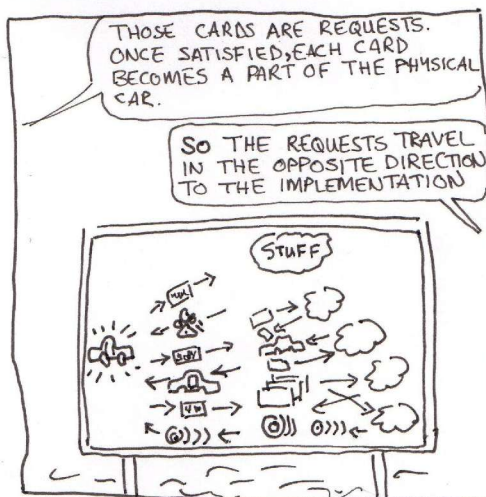
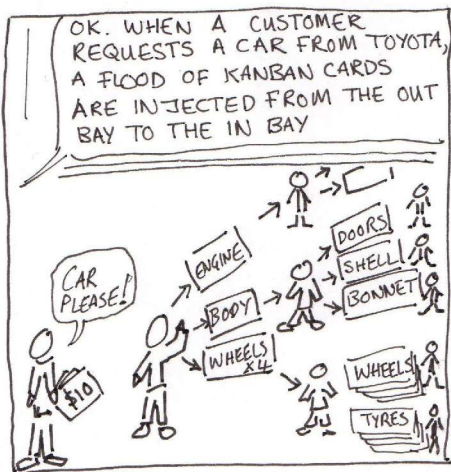
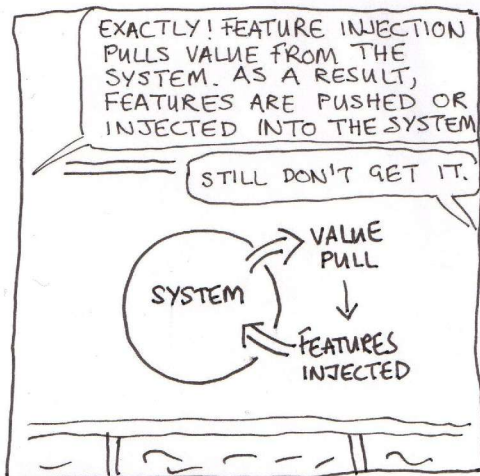
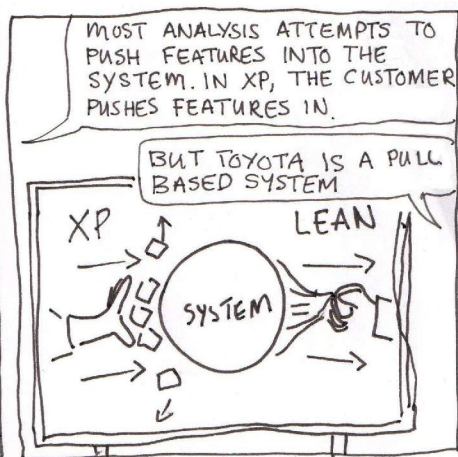
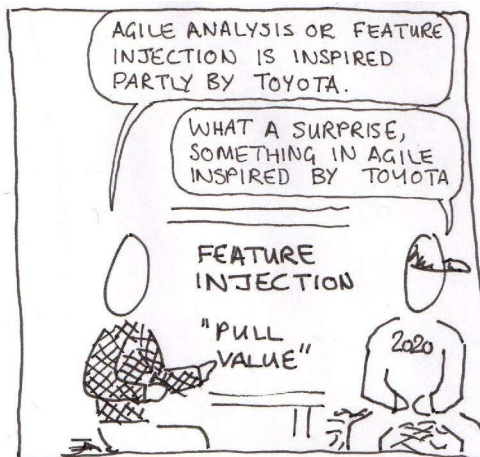
CUSTOMER DIFFERENTIATING	FIND A PARTNER	INVEST AND EXCEL
NON CUSTOMER DIFF	WHO CARES	GOOD ENOUGH
	NON MISSION CRITICAL	MISSION CRITICAL

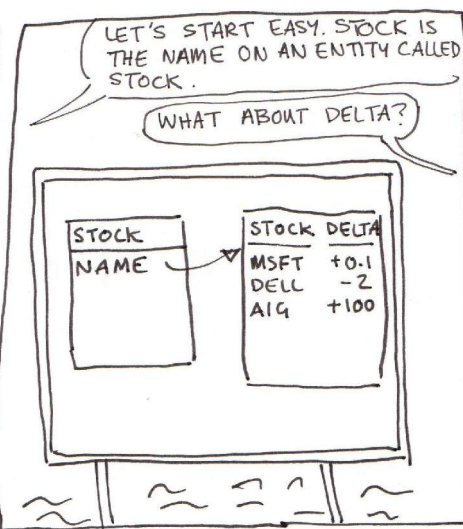
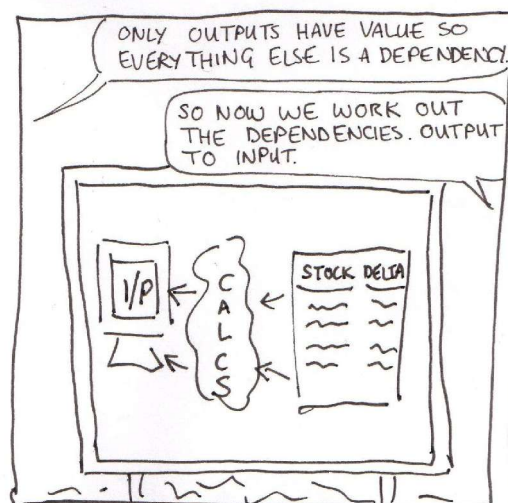
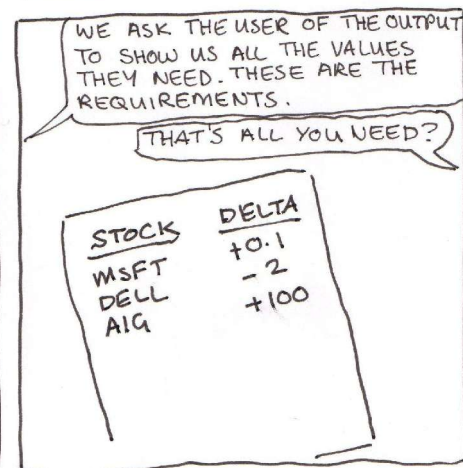
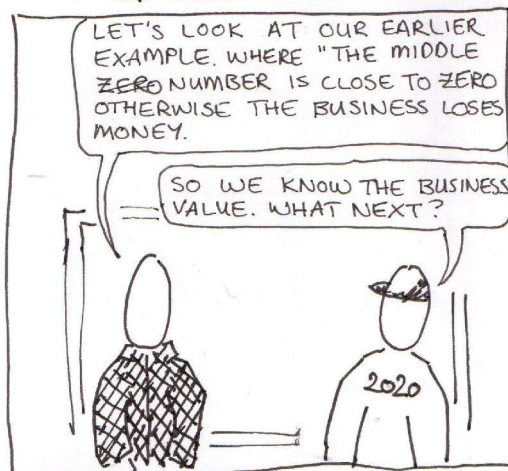
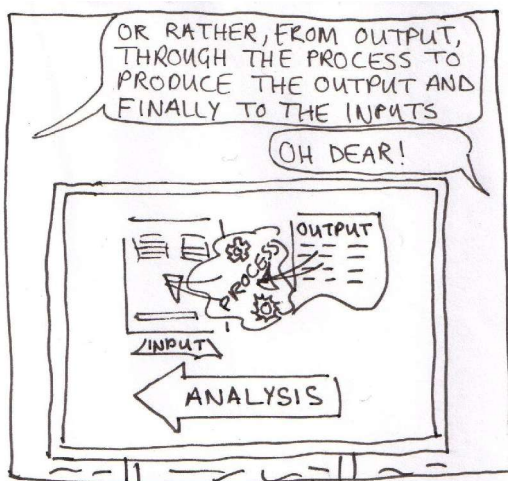


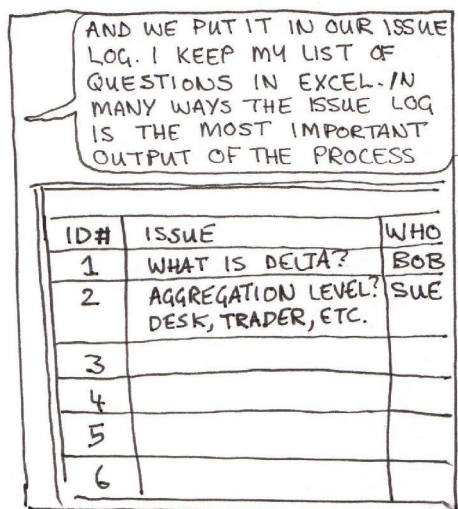
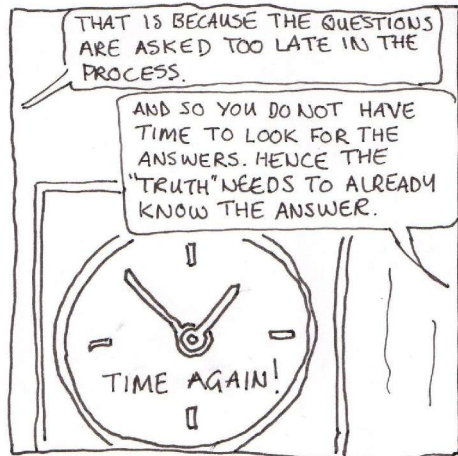
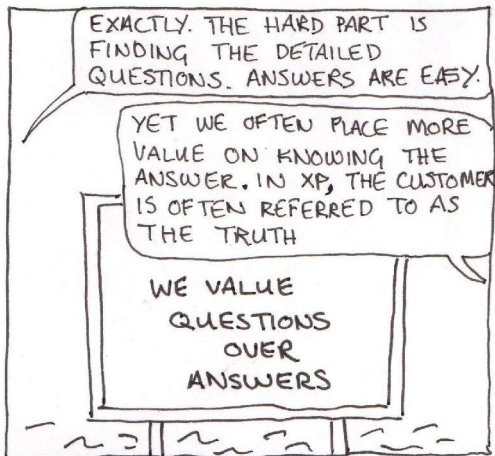
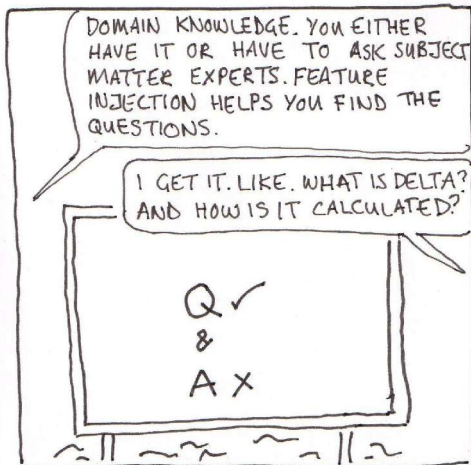
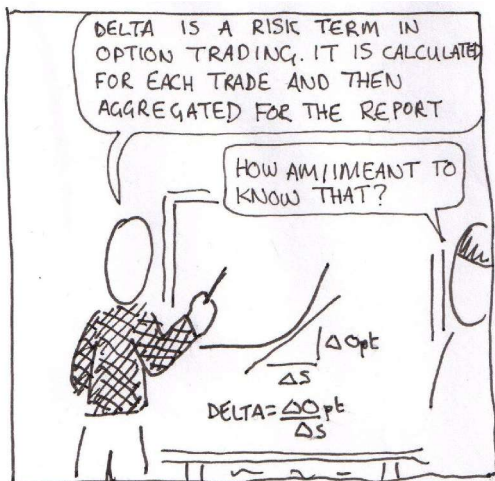


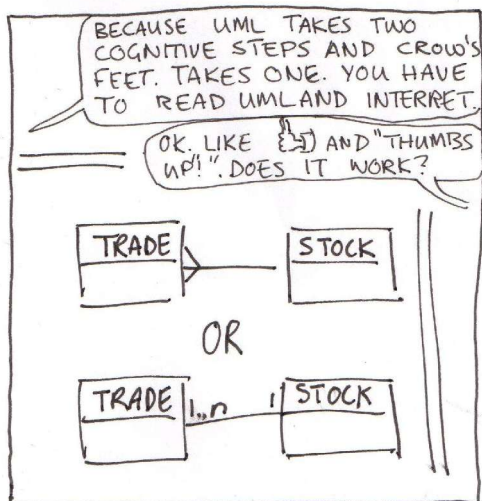
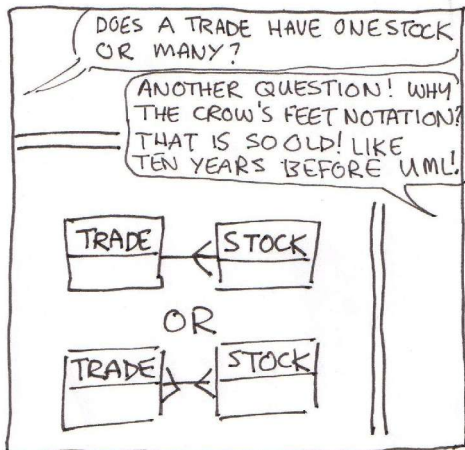
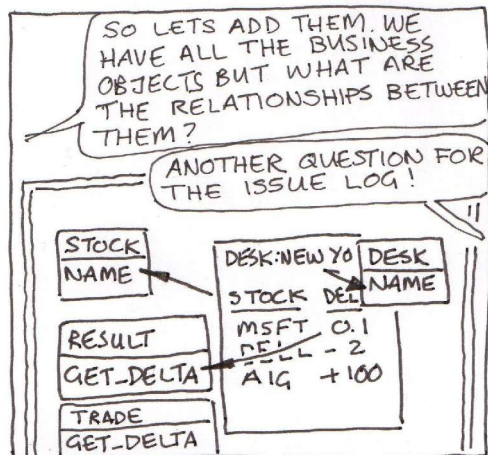
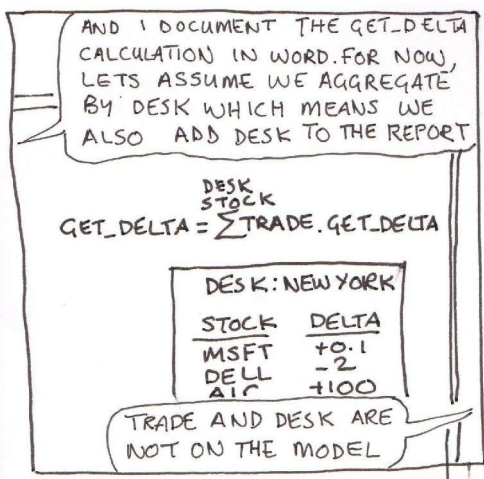
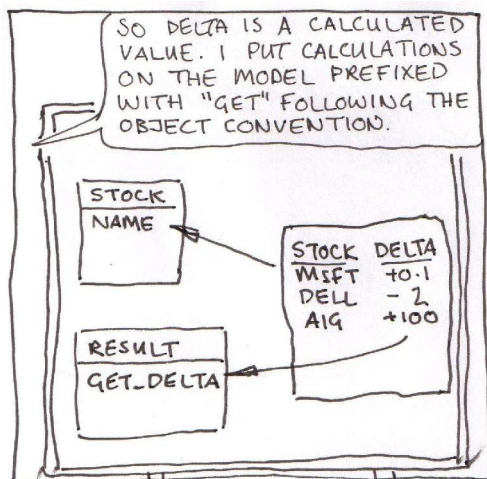
accelinnova.com

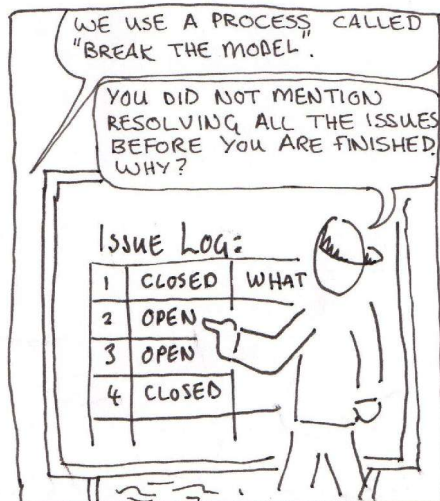
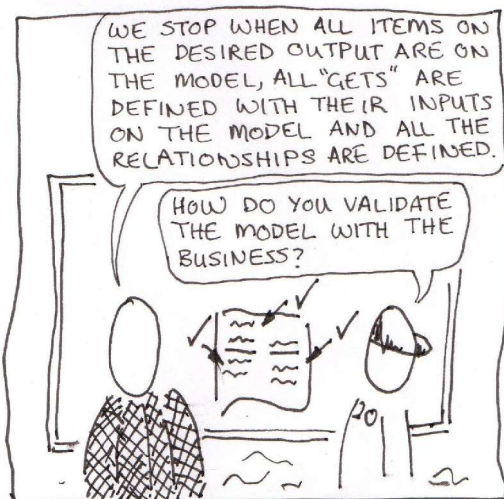
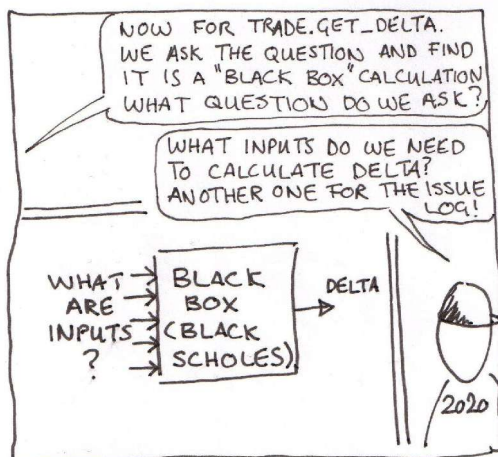
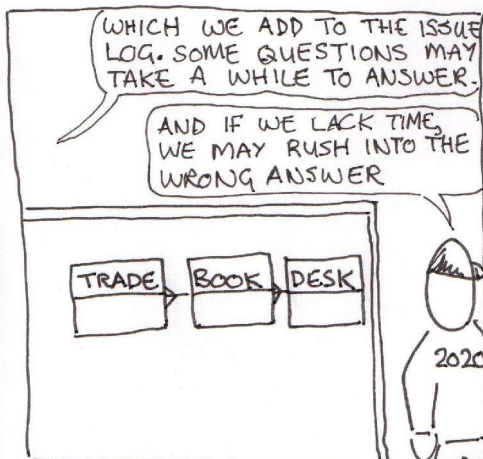
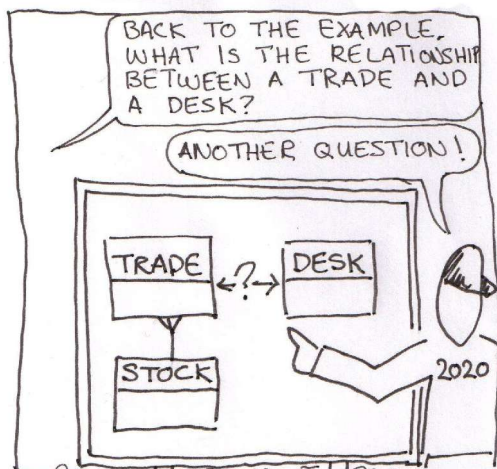


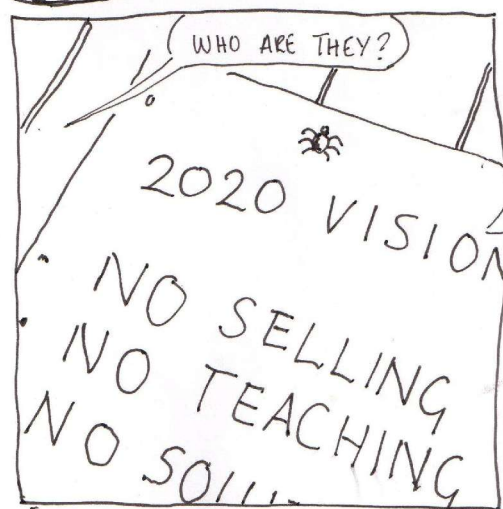
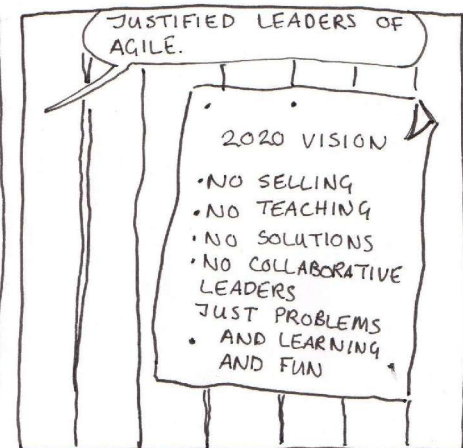
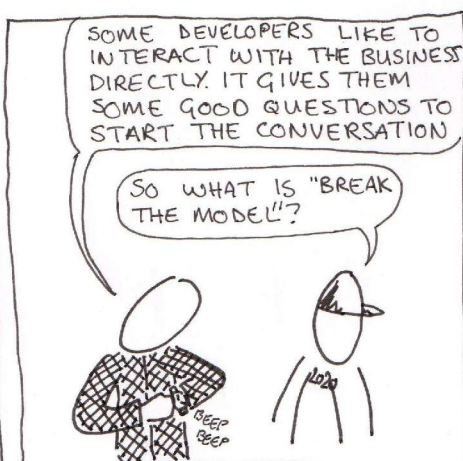


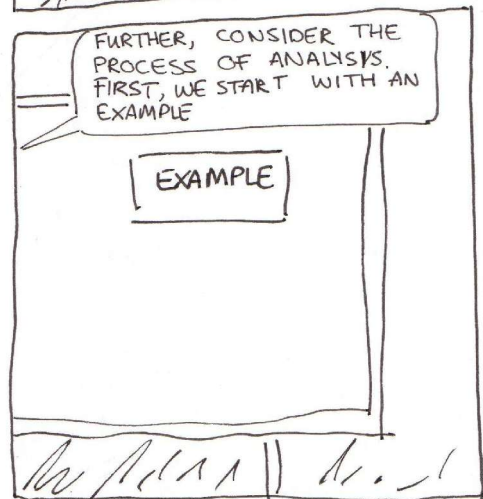
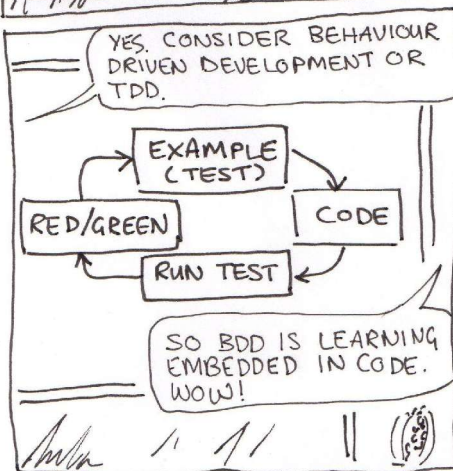
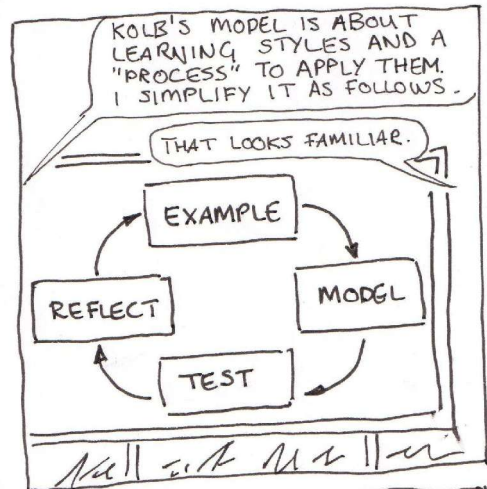
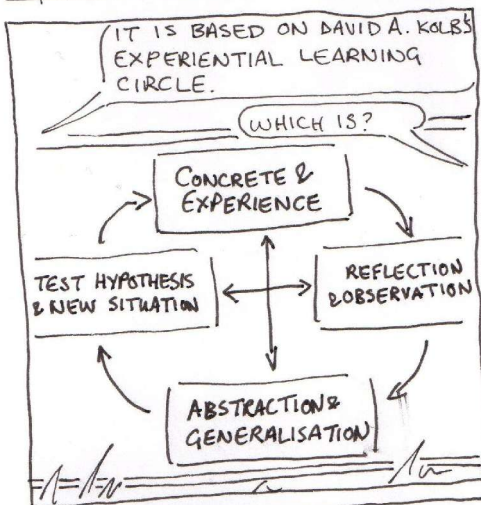
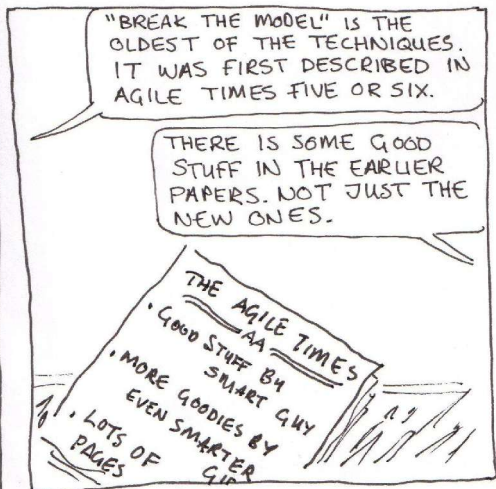


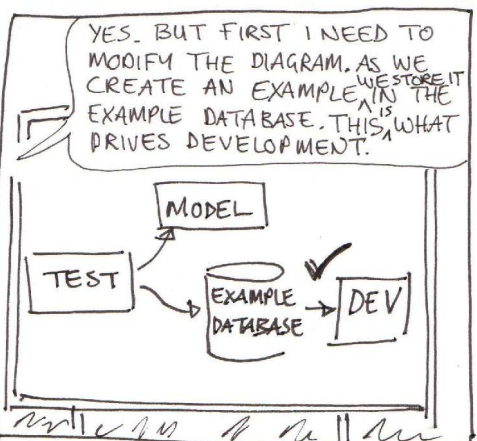
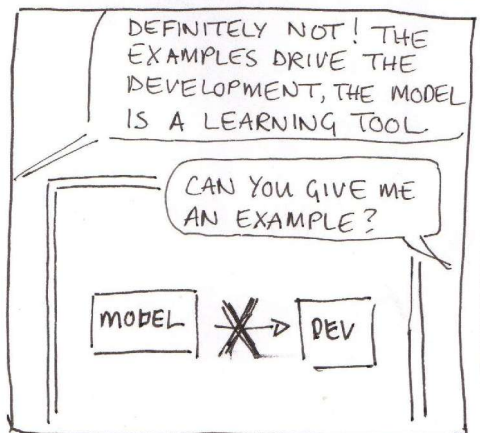
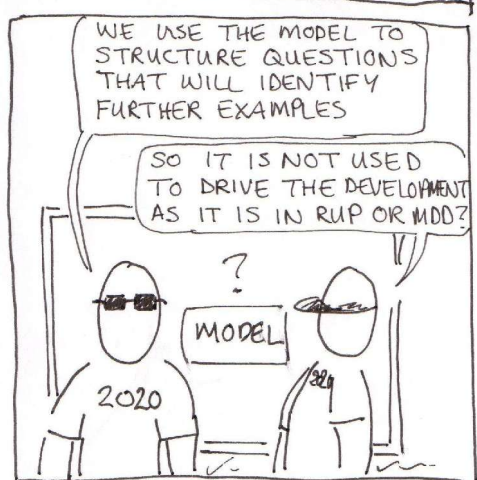
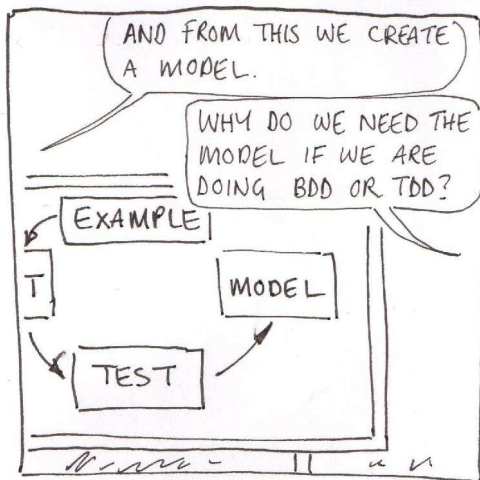
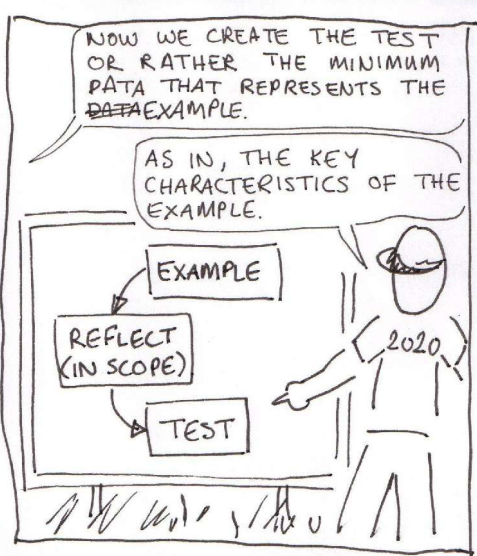
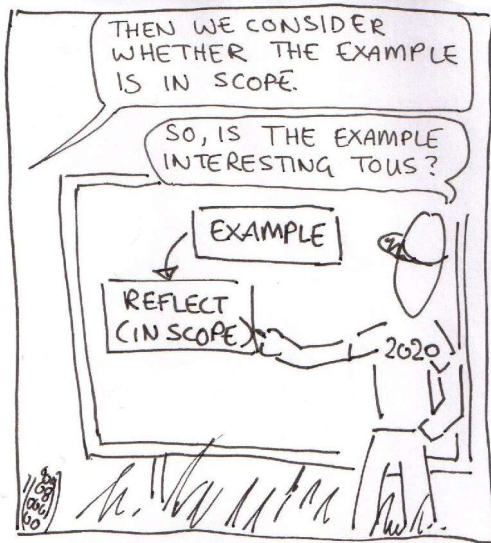


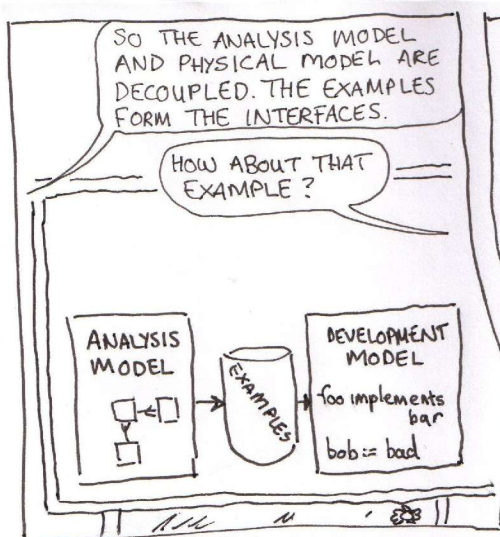








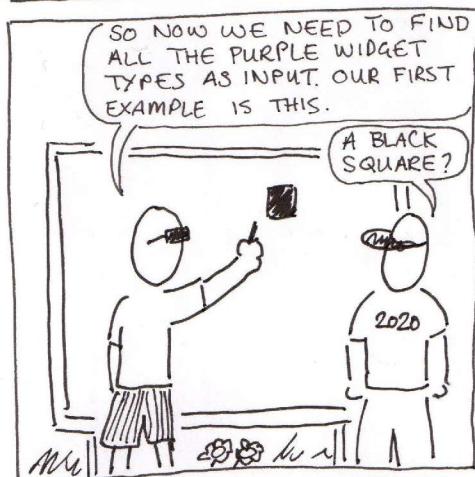




IMAGINE WE ARE IMPLEMENTING A PURPLE WIDGET SCORE REPORT. THE PURPLE WIDGET CALCULATION IS A BLACK BOX.

O.K. A BIT LIKE DELTA IN THE LAST EXAMPLE.

DEPARTMENT	P.W.S.
ENGINEERING	20%
H.R.	1%
SALES.	80%



FOR THIS EXAMPLE EXERCISE, LET US MARK SCOPE WITH A TICK OR CROSS WHICH WILL RANDOMLY CHOSE. SO OUR MODEL IS?

A BLACK SQUARE

EXAMPLES	MODEL
■ ✓	BLACK SQUARE

BLACK IS A COLOUR. SO WE ASK OUR BUSINESS. "DO YOU HAVE ANY SQUARES THAT ARE NOT BLACK. WHITE FOR EXAMPLE.

ANOTHER QUESTION FOR THE ISSUE LOG. AND OUR MODEL WOULD BE A BLACK OR WHITE SQUARE.

EXAMPLES	MODEL
■ ✓	BLACK SQUARE
□ ✓	B/W SQUARE

AND THE NEXT OBVIOUS QUESTION?

ANY NON BLACK OR WHITE SQUARES? RED, BLUE... ANY WOULD BREAK THE MODEL

EXAMPLES	MODEL
■ ✓	BLACK SQUARE
□ ✓	B/W SQUARE
▣ ✓	COLOURED SQUARE

SO NOW WE CAN ASK IF THERE ANY EXAMPLES THAT ARE NOT SQUARES.... TRIANGLE ETC.

SO THE DISCUSSION IS BASED ON CONCRETE EXAMPLES RATHER THAN ABSTRACT MODELS

EXAMPLES	MODEL
■ ✓	BLACK SQUARE
□ ✓	B/W SQUARE
▣ ✓	COLOURED SQUARE
△ ✓	STRAIGHT SIDED SHAPE
○ ✓	SMOOTH SIDED SHAPE
☁ ✗	COLOURED SHAPE

AND EACH EXAMPLE BREAKS THE MODEL, WHICH HELPS US FIND A MORE ENCOMPASSING ONE.

SO THE "BREAK THE MODEL" PROCESS HELPS US "CHUNK UP" AS EDWARD DE BONO CALLS IT.

```

graph LR
    C[COLOUR  
NAME  
PW_RATING] --> S[SHAPE  
SMOOTH?  
GET_PWS]
    S --> ST[SHAPE TYPE  
NAME  
PW_RATING]
    N[NUMBER  
NAME  
PW_RATING] --> SI[SIDES]
    SI --> T[TEXTURE  
NAME  
PW_RATING]
  
```

SO RATHER THAN PROTECT THE MODEL ...

WE HELP THE BUSINESS TO BREAK IT, GIVING THEM QUESTIONS THAT WILL BREAK IT.

WEAK POINT!
ONE OR MANY

IN FACT ON ONE PROJECT, WE OFFERED A BOTTLE OF CHAMPAGNE FOR EVERY EXAMPLE THAT BROKE THE MODEL.

WHAT HAPPENED?

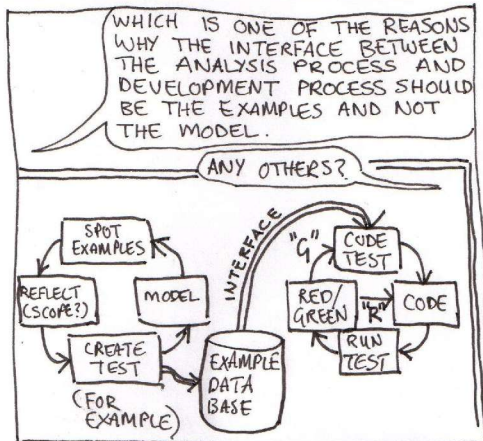
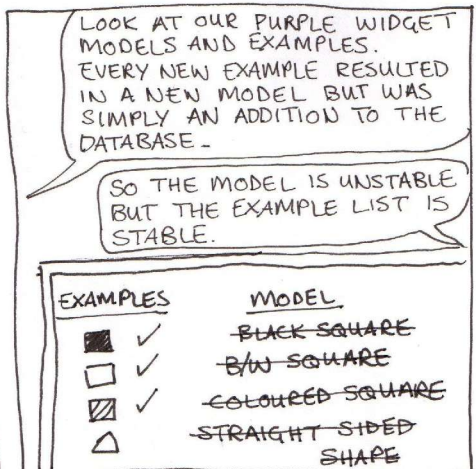
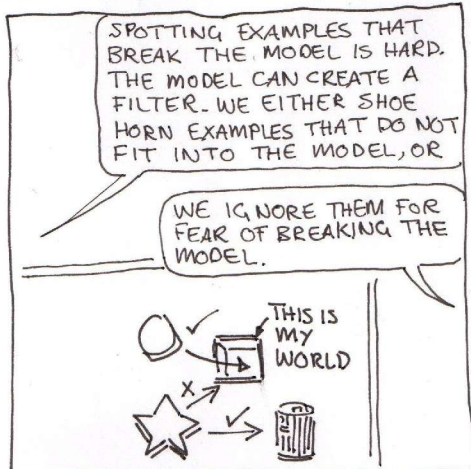
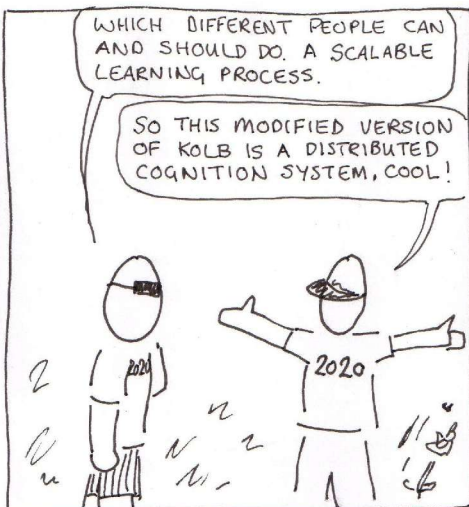
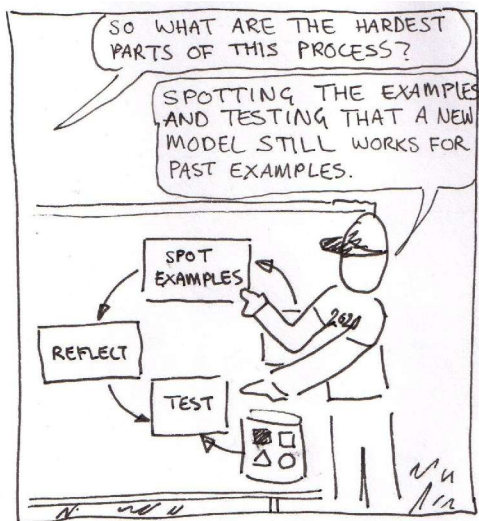
NO ONE BOTHERED. BUT THEN ONE GUY FOUND ONE AND WE PRESENT HIM WITH THE CHAMPAGNE AT A PROJECT MEETING.

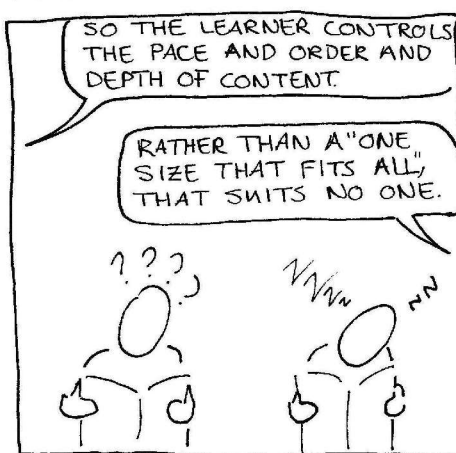
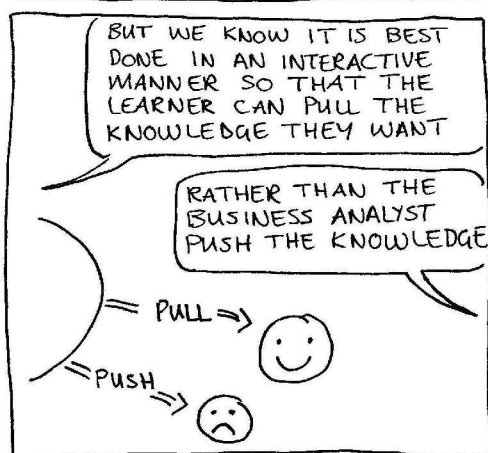
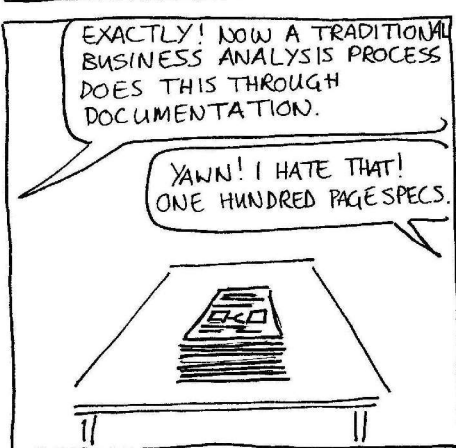
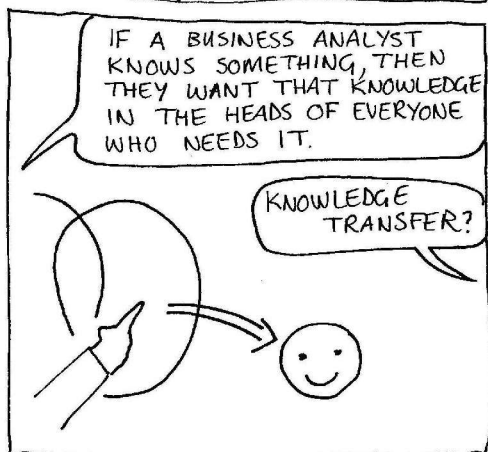
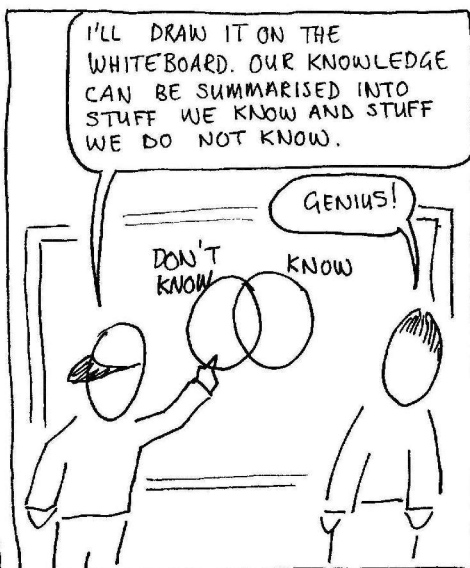
THEN WHAT?

THEY TORE THE DEPARTMENT APART LOOKING FOR EXAMPLES. AND WE HANDED OUT SIX MORE BOTTLES OF CHAMPAGNE.

I GET IT. STATUS!

IF HE CAN FIND ONE. I CAN FIND FIVE!





SO NOW THE BUSINESS ANALYST CAN WRITE THE SPEC. IN A WAY THAT THEY WILL NOT FORGET ANYTHING AND KNOW THE AUDIENCE THEY ARE WRITING FOR....

THEMSELVES.

SPEC
MY OWN
KNOWLEDGE
DATABASE

BACK TO OUR KNOW/DON'T KNOW DIAGRAM. NEXT WE HAVE STUFF WE KNOW WE DON'T KNOW

SO YOU KNOW THE QUESTION BUT NOT THE ANSWER.

SO ALL YOU NEED TO DO IS FIND THE ANSWERS, WHICH IS EASY IF YOU HAVE TIME.

BUT A NIGHTMARE IF YOU NEED AN ANSWER IMMEDIATELY.

IF THE ISSUE RESOLUTION PROCESS TAKES TOO LONG OR THE STAKE HOLDERS CANNOT AGREE, IT IS BETTER TO BUILD MORE THAN ONE SOLUTION AND DECIDE LATER

REAL OPTIONS!

DELTA IS $\frac{F(S+\Delta) - F(S-\Delta)}{2\Delta}$

NO, NO, NO! DELTA IS $\frac{F(S+\Delta) - F(S-\Delta)}{2\Delta}$

BACK TO THE DIAGRAM. NOW WE HAVE THE STUFF THAT WE DO NOT KNOW THAT WE DO NOT KNOW

SO WE DO NOT EVEN KNOW THE QUESTIONS! NASTY!

THIS IS WHERE FEATURE INJECTION COMES IN. IT IS A STRUCTURED META LEARNING PROCESS

WHY META?

"META" BECAUSE YOU CAN USE ANY MODELLING PROCESS YOU LIKE. UML, COLOUR MODELLING, MATHS, ANY YOU LIKE.

THAT'S COOL. BUT WHAT ABOUT RISK?

MODEL

- ~~BLACK SQUARE~~
- ~~B/W SQUARE~~
- ~~COLOURED SQUARE~~
- ~~COLOURED STRAIGHT-SIDED SHAPE~~

YOU TELL ME. WHERE IS THE MOST RISK?

WHEN YOU DO NOT KNOW THE QUESTION. WHEN YOU FIND THE QUESTION, IF THERE IS NOT ENOUGH TIME, YOU MAY HAVE TO BUILD MORE OR STOP WORK

SPOT ON. IF YOU KNOW THE QUESTION, YOU HAVE BOUNDED UNCERTAINTY THAT YOU CAN MANAGE.

SO BUSINESS ANALYSIS IS ~~RISK~~ RISK MANAGEMENT, AND THE ISSUE LOG IS THE MOST IMPORTANT OUTPUT FROM THE PROCESS

IF THE ANSWER IS 'A' THEN WE DO 'X' OTHERWISE WE DO 'Y'.

THAT'S ONE WAY OF LOOKING AT IT. NOW HOW MUCH TIME DO YOU SPEND ON EACH ACTIVITY

ABOUT 80% WRITING THE SPEC. 15% ON ISSUE RESOLUTION AND 5% ON FINDING QUESTIONS

SO HOW COULD YOU ADD MORE VALUE?

LESS TIME ON THE SPEC. AND MORE ON FINDING QUESTIONS.

GOTTA GO. I'M MEETING SOME OF THE FOLKS FROM THE REAL OPTIONS YAHOO GROUP TO DISCUSS LIQUIDITY... I THINK LEARNING HAS SOMETHING TO DO WITH IT.

R.O.S.E. SENT LETTERS

<p>Dear Olav,</p> <p>What is Chris's problem with XP and Scrum? He bashed both in this comic?</p> <p>Concerned, Detroit</p>	<p>documentation to prove that he a <i>Certifiable Sum Trainer</i>.</p> <p>Olav</p>	<p>Chris and I are working on creating a real options business graphic novel. The artist, Andy Belanger, drew the cover of this comic you're reading. You can see two sample pages on our blog at www.decision-coach.com.</p> <p>Also we are in negotiations with publishers to publish this real options book.</p> <p>Olav</p>
<p>Dear Concerned,</p> <p>Chris uses practices from both XP and Scrum. He considers XP practices on someone's resume to be a good indicator of a high quality candidate. He tweaks the nose of the XP and Scrum community to get them to attack the Feature Injection ideas. When they "Break the Model", he simply fixes it. The hardest thing after all is to see the problems with the model. Breaking the model is the ultimate form of feedback.</p> <p>Olav</p>	<p>Dear Olav,</p> <p>First you guys pretend Chris is dying of a coughing fit. Then Chris writes a comic book. You direct the Agile Fringe? What's next?</p> <p>Concerned, Seattle</p>	<p>Dear Olav</p> <p>Why comic strips and not a traditional article?</p> <p>Concerned, Sydney</p>
<p>Dear Olav,</p> <p>Is Chris a mad professor?</p> <p>Concerned, Des Moines</p>	<p>Dear Concerned,</p> <p>Lots. InfoQ has just published an article on Collaborative Leadership and Collaborative Management with Pollyanna Pixton. Chris is working on a comic book on real options and Theory of Constraints, and a short comic strip to prove Black Scholes should never ever be used for real options. Chris is also working to create a series of real option based conferences called 2020 vision. The idea is to start with a vision of what the world is like in the year 2020.... both good or bad. People then work out what they need to learn to make it happen or prevent it from happening. The aim is to identify gaps in our knowledge and build community around plugging those gaps rather than present solutions.</p>	<p>Dear Sydney</p> <p>Comics allow us to create an easy to follow dialog which makes it easier for you the reader to understand instead of the usual monologue in an article. (Some might say it's even easier than listening to Chris.) Also it allows us to answer the same question many ways rather than use the best we know. The comic is the easiest way to translate a presentation into a paper form. The slides becomes the pictures in the strip.</p> <p>Olav</p>
<p>Dear Concerned,</p> <p>Yes and no. Technically he is insane, but he prefers the term nutter. He does not like anyone calling him a loony because his moods are not affected by the phases of the moon, he is like it all the time. In addition, he helps others to learn financial maths. If you ever visit his home he has</p>		<p>See ya in tha bar!</p>

NOWTS

Please note that any marks on this page may reduce the value
of the comic in future years