

A colleague found this remark in one of my tutorials: "In my experience the 'zero defects' attitude results in 50% less defects almost overnight" and asked me to explain this to a project team he was coaching. I thought that my experience with Zero Defects might be interesting for more people than just this project team.



Niels Malotaux

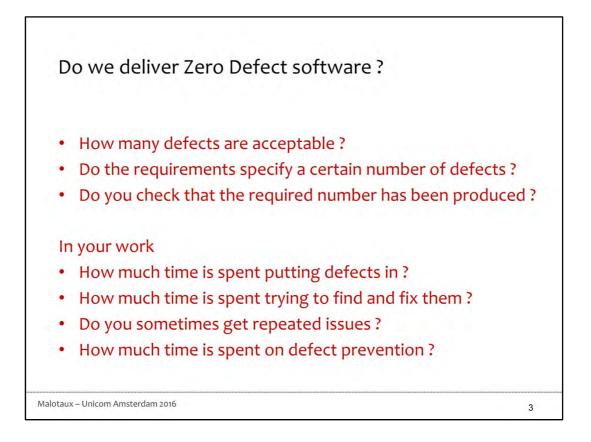
Graduated Electronics at Delft University of Technology in 1974

Army service at the Dutch Laboratory for Electronic Developments for the Armed Forces, designing computer systems

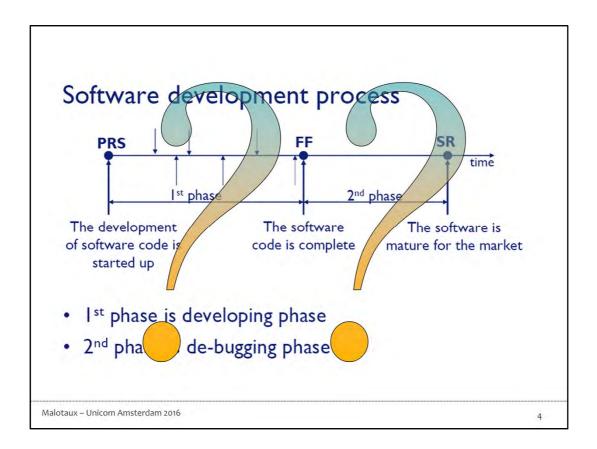
Philips Electronics – Application support for microcomputer systems design (1976-1980)

Malotaux - Electronic Systems Design - : developing electronic systems for clients products (1980-1998)

Now: **N R Malotaux - Consultancy:** coaching projects to deliver successfully and much faster (1998-)



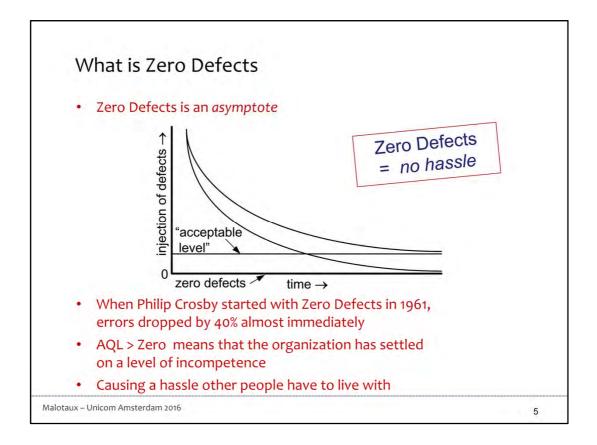
As many people think that even talking about ZD is useless, I'd like first to discuss some questions with the audience.



A University PhD student showed this picture as being the official development process at a well known large company in Holland. I've seen a similar picture in a presentation from a well known large software company in the US.

The 2^{nd} phase usually takes $50(\pm 30)\%$ of the total time. How can we call it "Code Complete" if it's full of issues?

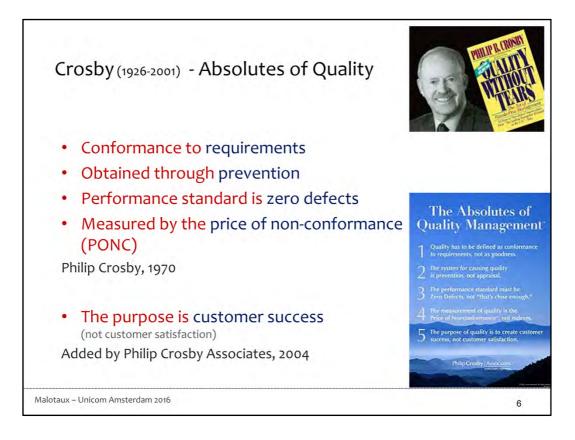
This is a very bad and costly process. However, because it's so widely practiced, many people think that this is how it should be. They should know better. Probably deficiency of the educational system, because the solution is known for decades.



When I actively started using the Zero Defects (ZD) concept in software projects, defects made decreased by at least 50% almost immediately. It took about 2 weeks before the developers understood that I was dead serious about it. Then the testers came to me saying: "Niels, something weird is going on: we don't find issues anymore! It simply works!" I said: "Isn't that exactly what we want to see? Now testing is becoming a real challenge, namely proving that there are no errors."

So, even if you don't believe that this can be true, if two people (Crosby and me) did it and showed a huge decrease of errors *made*, only by adopting the attitude, isn't it at least worth a try, especially if you realize that about half of most projects is spent on finding and fixing defects. That's a huge budget. Any savings on that is probably well worth trying.

"No Hassle" proved to be easier to use than ZD: Don't cause a hassle. No hassle to yourself, to your peers, to your organization, to your customers.



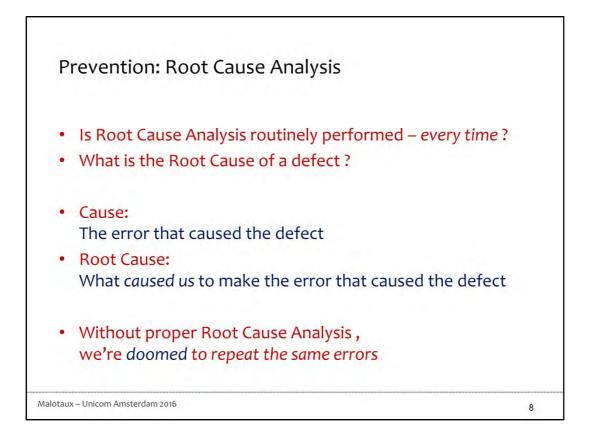
Philip Crosby defined the four 'Absolutes of Quality'. When I started as a coach in a company recently, I gave his book "Quality without tears" to the CEO for homework: "Next week I'll check that you read it!". He did and it immediately had an impact on his behaviour. He delayed a major release to first get rid of the hassles that we were going to deliver to the costumers. He also calculated the 'Price of Non-Conformance' (PONC), to be at least a quarter of a million Euro in the past year.

Phil Crosby's organization later added a 5th Absolute: Customer Success. I agree completely. But I don't agree with them adding "... not customer satisfaction". After all, the customer should be successful, but satisfied as well, as we'll see on the next slide.



This is to me the top-level requirement for any project or any work we do.

- The customer is the entity that orders and pays. The customer, however, in many cases doesn't use the result of our project himself. He gets the benefit through the users of the result.
- · What the customer says he wants is usually not what he really needs
- The time he needs it may be earlier or later than he says
- If the customer isn't satisfied, he doesn't want to pay
- If the customer isn't successful with what we deliver, he cannot pay
- If he's not more successful, why would he pay?
- What the customer wants, he cannot afford. If we try to satisfy all customer's wishes, we'll probably fail from the beginning. We can do great things, given unlimited time and money. But neither the customer nor we have unlimited time and money. Therefore: The requirements are what the Stakeholders require, but for a project: the requirements are what the project is planning to satisfy.
- The customer is king, but we aren't slaves. Both sides should benefit and be happy with the result.
- We will get the best result in the shortest possible time, but not shorter than possible. The impossible takes too much time.



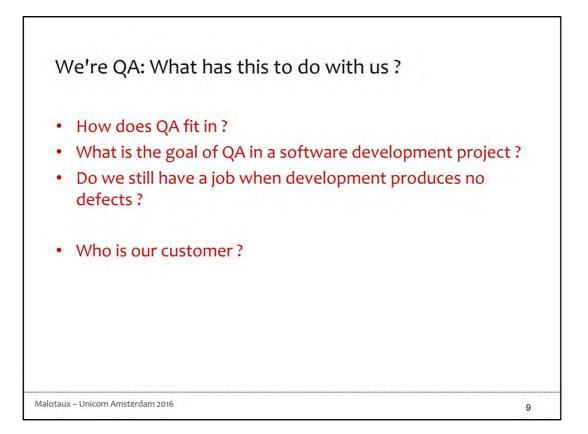
Years ago I suggested to add a box for the 'Root Cause' and for the 'Root Cause Suggested Solution' in a bug-tracking system. When I later checked how people were using this, I found that in the Root Cause box they documented the cause of the bug and in the Root Cause Suggested Solution box the suggestion how to repair the bug.

Apparently, they didn't see the difference between 'Cause' and 'Root Cause':

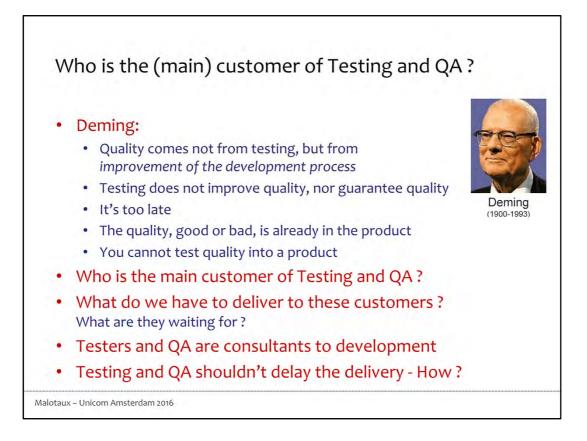
- The Cause of a defect is the error that caused the defect
- The Root Cause is what caused **us** to make the error that caused the defect

In another project I asked the project manager what they do with the results of the code reviews. "People repair the bugs" he said. I asked: "Don't you do Root Cause Analysis, in order to learn how to prevent this type of error from now on?" The response was: "On every issue we found??? We have no time for that!"

Apparently they have no time to learn to prevent, and rather spend a lot of time to find and fix(?). No wonder that projects take more time than they hoped for.

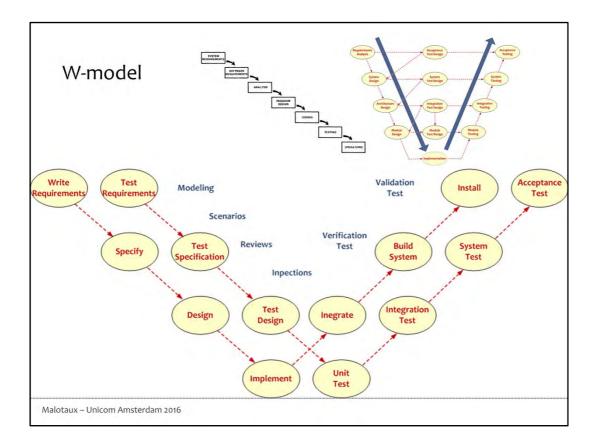


Many people equate QA with Testing. Testing, however, is just one of the quality measuring instruments of QA and hence only a small part of QA. So, let's shortly discuss what QA actually is and who the customer of QA is.



I experienced that to most testers this quote from Deming is quite a paradigm shift and usually comes as a shock. But usually it's a shock of recognition! It will change their attitude for the better forever.

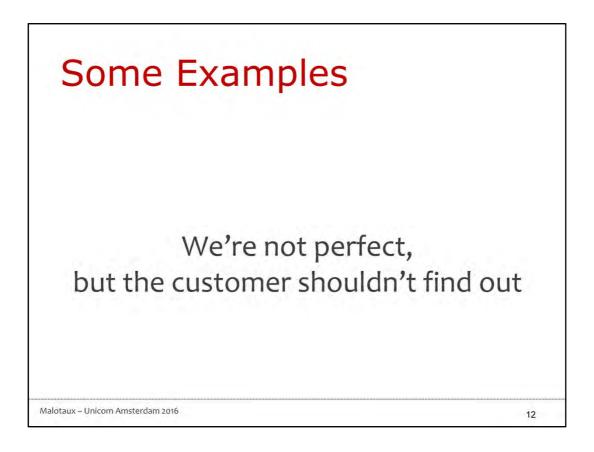
Now let's see how we can optimize our contribution as consultants to development.



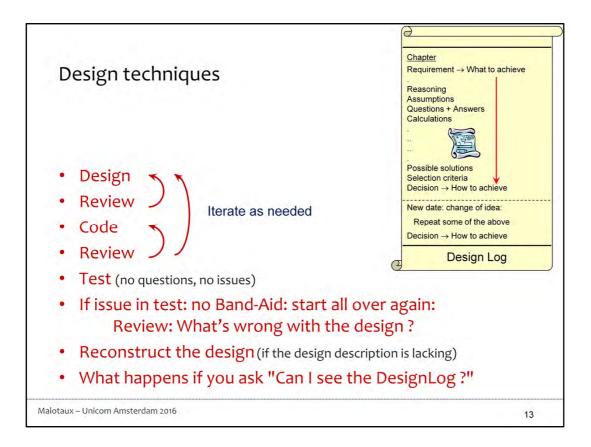
In the Agile world, the Waterfall and derived models are often seen as bad. However, these models are still valid for every sprint. After all, we have to determine what value we should deliver (requirement), how we can and are going to realize it (design), how we implement it (coding), integrate it, test it, and deliver. For QA the challenge is not to test only the code and find bugs, but to help development to prevent the bugs in the first place. By reviewing the requirement, the design and the implementation, so that the final test can conclude that it simply works as it is supposed to work.

The V-model is actually a folded Waterfall, where the most expensive issues (requirements issues) are found at the end. The W-model shows that we should find the issues once they are created, rather than when they cause trouble later. All models are wrong, some are useful. If they're useful, we may use them.

The requirement, the design, the code: they're all different manifestations of the same product. However, only the code can be run to check that it 'physically' does what it is supposed to do. That's what we usually call 'testing'. Before we have code, there are other techniques to check that the product is right, like Modelling, Scenarios, Reviewing, Inspecting. In these areas QA can prove its value as well.



Let's discuss some examples of techniques that helped me and others to move towards Zero Defect deliveries.



There are many techniques known to approach ZD faster. One of them is what I call the DesignLog.

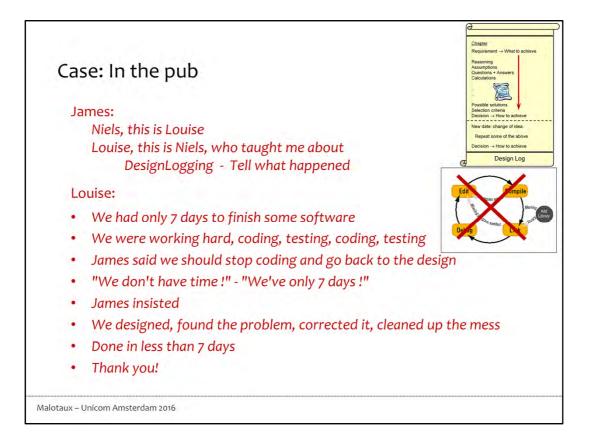
When I started my career at Philips Electronics in 1976 (at the same time Philips started to sell its first microprocessor), we got a notebook to note our thoughts, experiments and findings chronologically. It was difficult, however, to retrieve an idea I had several weeks before, because it was buried in many pages of hardly readable handwriting.

Nowadays we can use a word processor, add pictures, organize by subject rather than chronologically, and search through the text. We log our thoughts in chapters, which start with what we have to achieve (requirement), end with how we think we will achieve it (implementation specification), with in between the reasoning, assumptions, questions and answers, possible solutions, decision criteria and the selected solution (design).

If I see design documentation, this usually only shows what people decided to implement, rather than also recording why and how they arrived at this decision.

The DesignLog should be reviewed to find possible issues before we start the implementation. Because the choices and design are well documented, in the maintenance phase (often a the largest portion of the cost of deployment of software!) minimum time is lost. One of the requirements for the DesignLog is: "If someone has to change something in the software one year later, he should be up and running within one or at most two days."

When QA asks development to review the DesignLog, if there is one they can review and also use this information to define and optimize their test-cases. If there is none, this is a good time to introduce the concept. See next slide.



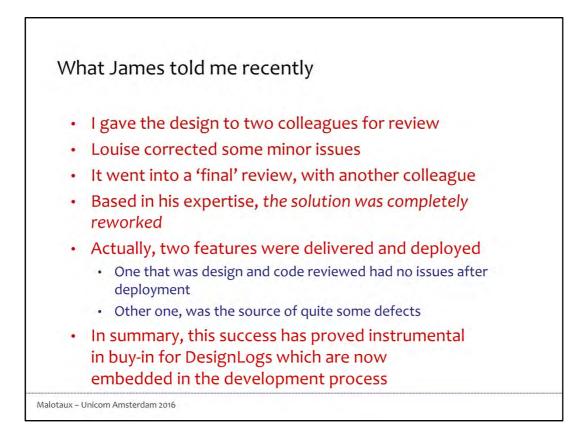
This case happened just a few months ago, see the text on the slide.

It's always nice to experience that the techniques that worked for me and for many others in the past, still work today. Many old techniques never get out of date.

We see, however, that it's not so easy to convince people to do something that seems counterintuitive: going back to the design rather than grinding on in code and leaving a lot of dangerous scars in the process.

Delivering quality often needs counter-intuitive measures.

On the next slide we'll see how a review even caused the whole design to be changed!

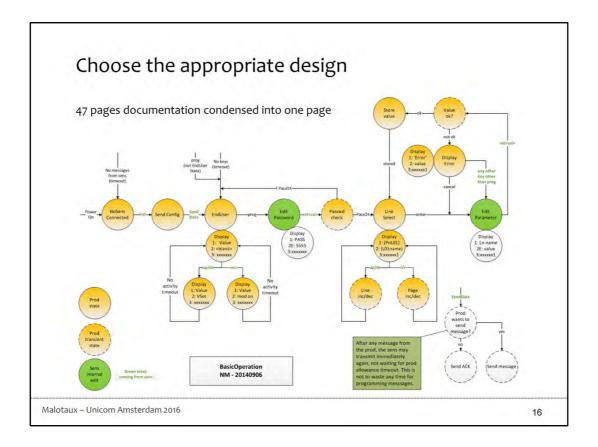


James explained me later more interesting details of this case:

- There were two features required for a release, one of which was on the critical path and placed the delivery at risk
- I saw an "opportunity" for a Design "prevention, rather than fixing" and also an opportunity to encourage documentation
- Because Louise struggled a bit with the design (not many people in software have been educated in how to design), we Timeboxed the initial draft
- I emailed it to two colleagues to review: "please review, assuming you will code-review the implementation, and based on this DesignLog you know what the implementation you will have to code-review will be"
- Louise emailed me in a panic that if she knew it would be reviewed, she would have written something different. I said - "no, do nothing yet - review first and then update with your new understanding and feedback. Reason: the next draft will be better."
- For the next review, another colleague who was not previously available was invited. At the meeting where I expected the DesignLog to be approved with minor modifications and to get estimates for the work involved, the Design was *totally reworked*
- We agreed the new Design was better than the original ideas
- Actually, two features were delivered and deployed
 - The one that was designed, reviewed, coded, and code reviewed had no issues after deployment
 - Another one, which was done in the 'traditional' way, was the source of quite a few defects
- In summary, this success has proved instrumental in buy-in for DesignLogs which are now embedded in the development process

Using Inspection (Review) in various ways:

- The review caused them not to implement a bad solution
- If they would have implemented the original solution, they probably wouldn't have found out until much later
- The whole process allowed them to deliver well before the deadline rather than after.



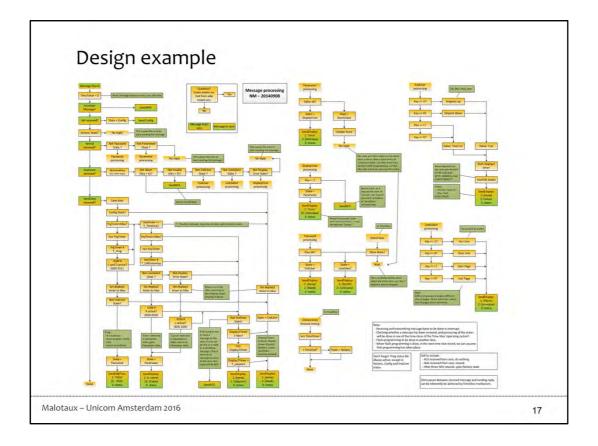
If I see documentation at all, it is usually just text. Sometimes a lot of text. One of my mantra's is: "Where are the pictures?"

This and the next slide are an example of some design I made recently (anonymised). You don't have to check the text and what it actually does. It's just to show some examples of concisely documenting functionality in a way most people, with a bit of understanding, can follow immediately.

This slide shows a design of the communication between some controller and a remote user interface. It was documented in a 47 page document by an 'architect'. 47 pages of interface description is almost impossible to oversee by humans, hence it contained a lot of inconsistencies and the people who had to implement it actually ignored it.

Once I made this one page overview, we could discuss, ease out the inconsistencies, make decisions, agree, and everyone knew exactly what to do. Conclusion: just documenting isn't enough. We have to learn how to document for usefulness.

QA can ask a developer to explain how the interface should work. If the developer only shows code to review, we know we have a problem. If the QA person doesn't understand the explanation, the explanation apparently isn't clear enough, which is a big risk for the quality of the result. If it's only text, it won't work either.

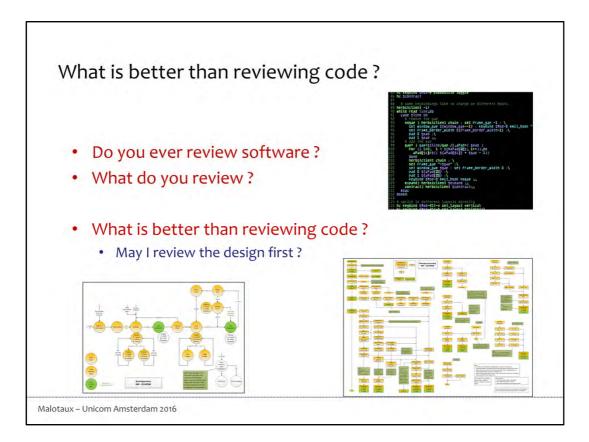


This is how I implemented the communication design based on my discussions with the suppliers of the remote user interface. The design was made to be reviewed and then it could readily be implemented based on this design. If I see how much I moved and reshuffled before I was content that this was right, I cannot imagine how this could be done properly in code without having this design. Like in the Cleanroom Approach to Software Development I designed down to a level of some 3 lines of code per design element. Sorry, I have no time now to go into detail, but the Cleanroom Approach routinely delivered an order of magnitude less defects in shorter time. Making changes in the code is not allowed before we have updated the design. The code should always be derived from the current design. Reviews of code should always check that the code does what the design says

These were just examples. The challenge is every time again to find the right representation that is easiest to comprehend and review.

Of course the projects the audience is working in usually do these things properly. But I still see too often that the 'design' is only in the mind of the developer who writes the code, or just a rough sketch, with devastating effects in software quality and delivery time.

If as a QA person you encounter these effects, think what you should do about it.



Many people do reviews, but have they been trained in reviewing and what is the best way to optimize the effectiveness.

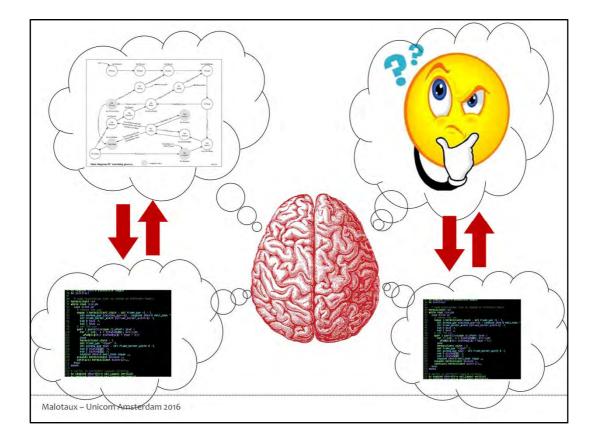
Don't waste your time on reviewing code if you haven't reviewed the design. After the review of the design, the code will be changed anyway, so what's the point of first reviewing the code?

If there is only code and no design, reconstruct the design. If you work only in code, producing bugs is to be expected. Bugs shouldn't be there. Hence, don't work only in code.

inducts	1-1/bir/besh	1 Willingtash- E-W safe a Dustingth Ver as Laser -		and
and morth has easy pictures per year	8 0 Check +1 +/, can/ig/merbillu/tak/* +/bin/* +/, theses/urecources-gen confinently, can/ig/merbillu/tak* +311/11 Aresetclient			Testo Casting States (CAST AND CAST
a Magnada (C. 1011) (M. 11) (M. 11) (M. 11) Magnada (C. 1011) (M. 11) (M. 11) (M. 11)	s within mericial acti C associations "10", 1 association acti C associations "10", 1 and C comparisons (10, 10", 1	<pre>> fice*11" { _s *28"] & boales*28" scales280 sames*echo *15"/cet -6", *-F1"</pre>	Starte Start, denurc	sleepless" pidmi/sleepless"
		convert -scale Socale "pric" "State"-thush.org		I lossie() ([# faleerless] 44 re ssteerless II touch saleerless
fer fer haltallte lint			<pre>> Pdts:/pdts.tpdt/dis* L0 cachedirs/pdtb 2004/dis* L1 f = fccchedirs* 1, then L2 cachedicachedir/dees_fun L2 cachedirachedir/dees_fun</pre>	in annel (
<pre>case -gi 1] 66 echs "der/ca heefs"wicht((snam - 1)).html">C presi case "der/" >> piccripyar".html page -ee stationage] 66 echs " ca heefs"wicht.html">C piccripyar nam -ee stationage] 66 echs " ca heefs"wicht.html">C piccripyar nam - ca heefstationage ; 66 echs " ca heefs"wicht.html">C piccripyar nam - ca heefstationage ; 66 echs " ca heefstwicht.html">C piccripyar nam - ca heefstationage ; 66 echs " ca heefstwicht.html">C piccripyar nam - ca heefstationage ; 66 echs " ca heefstwicht.html">C piccripyar nam - ca heefstationage ; 66 echs = ca heefstwicht.html">C piccripyar nam - ca heefstwicht.html >> piccripyar = ca heefstwicht.</pre>	Interestical -/ Interest Construction -/ Interest Participation -/ I a	18 41 2. 6 ela Filitziaeran	1 13 else 1 19 cachespubly exemujcache il 10 ha sug birs, vatt baux to bourity i	S [+ * * (sischarge *] 66 [+e "S(pig eplayer)"] 66 [+ ***
	he set frame burder active color stframe burder active color-bfg.cl	2 wikes strast	126 127 cedist	1003
	<pre># No Get Frame_be_bermai_color s(Frame_be_bermai_color-mododob) % No Get Frame_be_brameparent s(Frame_be_transparent-1)</pre>	num-configs() (11 Print and a Theorem Hills man	<pre>bill (able f street) -st 5 100 -s "((dissary)") 1 00 terrer to street) -st 5 100 -s "((dissary)") 1 00 terrer to street) terrer to street (street) -st " terrer to street) terrer to street terrer to street terrerer to street terrer to street terrer to street</pre>
that cohard pages with Stiest wick ofter three at the and better at in Alexand and an and an and an and better better the actual three large	he set einden berder active colar Steinden berder active jolar-frg	Builtin of ShiT Bil off "" Bil connil right salter	ette "Austatest "ta shate i sert -u/asits)"i tee "scach	entityaunting text "hattery very tes."
nes week" stier" beer nes week" stier" beer conter ())	to set default_frame_layout 3 to set focus_dilust_shift 1 to set facus_dilust_shift 1 to set facus_dilust_frame 0 to set facus_dilust_frame 0 to set facus_dilust_frame 1		The presence of "scalar" share in series of the "scalar" scalar is series - on the scalar scalar is series - on the scalar	the steep 10s
saut(**)) 1 martials, and class the wags. 1 foount age plikeit 66 (10 Page+))	hc set raise an rocus 1 hc set always show frame 0	33 atlas st'sude'		21 ga
		<pre>is allow allowingColland, SHARLY, ashed" allow viewed="state"</pre>	14 tops fors to fors in . 33 to cs "stat" to catemara.	14/witten date hit
stes off the last ever	he rewarded	The second secon	10 case "Scent" in 27	2 E File browling will incom
	The Replind Control-All-MackSpace with the keybind Shor-Shifter reload inc keybind Shor-Shifter close in keybind Shor-Shifter close	<pre>alsas http://doi/ou//alsa/alsa/alsa/alsa/alsa/alsa/alsa/al</pre>	20. there 3 I charact the union scheme	Source SHORE/.menurc
	hc keybind stor-shirt-c close The keybind stor-x close	a paramanant	 Here) I Charges the union scheme scheme Here's (1) solid/, here's scheme scheme	A P DECENTS(DECKS-DECKS) -1 40"
33,2 41	n alles ne keybind (Rod-Return seam urunte	<pre>Files picture poces -5" files picture picture -5"</pre>	127 attaints er) a nations for artecting films	9 # FLITCLOF FAFE
*) NR 918. 18142 Tion: In the difference in the same meaning ciling wetherwork. Joint "weighter" (Derivitianing ciling americani, con- tract and same ciling and ciling americani.	n (Frieder) Ec Kegban Sho-Feleren spaan urusic Ec Kegban Sho-Feleren spaan surusic Ec Kegban Sho-Feleren Short/Takija Ec Kegban Sho-Feleren Short/Takija Ec Kegban Short-Period Short Sciences Statistic Spaan Short/Dakija Ec Kegban Short-Period Short Sciences Statistic Spaan Statistic		The second state of the se	12 cose "gat" in 13 cose "gat" in 14 cose "gat" in 15 avin amoin avin's funit, gatt, avin's, while, avin's 15 aving of the second s
e.+	be Regbine Rade-perine ipaan teanf/imap thee Spaan provie	attas gregatingasurt -spar	<pre>1</pre>	1) arlager "Set" 6 4
(a) when client in and out of "wapter" or "kines " in thema is any or walk" splits in County exploring in Arthonica water and the term of begins of the sector splits of the term of the term of the term.	 CONCRETE ALL AND ALL AND	Aliss gift" art-grt inslati Alia gov" art-grt inslati Alia gov" art-grt inslati Alia governer Alia governer	1 % is a line tolerictive program to a (erminal) 1 % and 1 94 whereast	A denter a macin 6
	he keybind lifeldudieleuservolume spann paninerincrease 5	alias ander's ant-net dist-ungrade" alias anaci's ant-net autoclean"	107 (If St. 1) 10 Find 1(sampath))1	
Trains shart and runs instant, good wapper for herboat. In beginn the rest instant /film/filmer state.press herb	2 Be Keybind Iffeludioffurte spann plainertoggie-mute be keybind Iffeludiofflag spann ape toggie be Weybind Iffeludiofflag togann ane toggie	alias anti's add-ant-repetitory'	(254***) 5 file (149420))) 15 ptd - (149420))) 15 ptd - (149420)) 16 ptd - (149420)) 17 ptd - (1494200)) 17 ptd -	sing state - s
Gier Devin Inte Cliente in Habier by Elack and runs commune, yestible erappe Homogrami Int Neglishet Bhol-Lemma spawn /PETV/10/14420 Habier Jahan Desnu,/or 1	Inc Keybind STdidudisPrev Spawn Mpc prev 5 hc Keybind STdiPawerd/F Spawn Shint/bin/pwr	attas grob-sedateriseda grob-ekceerig -a /beat/grob/grob.crg*	11 president vice of a loss that is in the loss that the set in a set in concernent in a set in the set of the	*. detis. odeis. odeis. dec)
iste tegener pro-tone in store, it is restared by a client of	A # 1412	Alies reports"sude shutteen of new"	1 57 uravi -e SSELL -c "Scer"	A ANALIA ANA)
1999 D' Client 1A aparte d'arte la craset, il la crefaced by a client d' laca IF alach in Edory, 11's remark, unet the film client can Ma beging the-lack team film film client client	0 Tes[HEVEn((E3) '4 '#)	alies rebosts sudo shuldown -r row" Alies Arits Arits buildown -h new" Alies busi usig shuldown -h new"	50 *) rete "tekt"15(SHLL-"/bis/ter") 6	it will will anterlives assaultant, men vie. 14 tas dire
	Decrement of the second	A File Star Star Star Star Star Star Star Star	50 953 #44	1 A Versener 2 a 2001, pr 10 roda
it patter Beternings splitt's itsatten value beteren 6.5 ann 0.6 i b R. Legist - statt's him tagent "mortagetal" vertical "mar myrti b	Argung(Thi, Athenses) No and schol Andricess	alaa tele'te -te	192	entrance - a tal -e reade via -e scott/.viare "statut"
direction"Beigentat" /attes: 0.442" Lagout"Yerticat" /ramot	he keybind that this wert index ti	T alsas dell'de -b"	2 et necempainte backspacenindent est start hillorint5 newrap ken y et reter sheeped incearin the separatus tabibers windenisting i et ne conservice viberou silemetristingers full t giskuperfabberaultCamptetionType = "content"	
Appute vertical" franci	d deal A mac veryfiad SALE-e Uraba - Apt eeb - Nor bed - Speen Firdran - V D W - yeyblan SALE-e war eeb - beynand SALE-shart-e euwr een	alles detrind is inset	<pre>5 et nu cursorline wildwenu wildwoderlist:tangest.rult 1 t et g.Superfabber.ultCompletionType = "context"</pre>	1 56
(herbitcliest SD.), bcc() (bc chain SD.); chn() ; cass="scase	the keybing that reason , keybing that space cycle_layout 1	i dias sho"mbil" 1 dias sis"autosus teggir" 1 dias asto"s supst"	<pre># Sletype playin indext on # set presidents, Money 10 gos</pre>	as a real Array to but the margine contents on people vacturation
	The region finance reasons , regions phone-space cycle_layout 1 the regions finance split within a finance is a the regions finance split whether a finance is a the regions finance split whether a finance is a	atian than then attach of more a minus will be build there be	10 4 northwigranning") i calarachean aalakai 10 dir i calarachean chera	and a sealing have naisered samplies the othe first and name.
The solution of the solution o	for L in (1. 1), 60 A subind shor-dit-ji selit merimental 0.51	ation than them attach of " doors a atoms will fee tool one too ation adartain -comine graphe.com" ation perform graphe.com ation atom attaches.com	133 mdga 137 mdga 138 mdga 139	W States in printy of a print of the of second within the second seco
The second secon	A de fili ante president der i is (1, 7), 60 en engelen descellen, soll artiket 5, 5 Ber angelen descellen, soll artiket 5, 5 Ber angelen descellen, soll artiket 5, 5 bei angelen descellen des beiden bei angelen descellen des beiden beiden des beiden des beiden beiden des beiden des beiden beiden des beiden beiden beiden des beiden	alsas tel"transmission-darmon" alsas tal"transmission-remote -a"	<pre>1 mid</pre>	if It s(PAD/v/hant/v/hant) i /hant 31,then
1115(dump[1044mm[10]-10484],[hestyls[10](1-11] 1241115(dump[104-),[metydl15(tertyl11-11)	The Regind Statestate cycle_layout +1 The Regind Statest Floating toggle		autoced buffrad /tep/ambi.edit* source /home/ninjaaron/.vim/pen/	if if it show + hidden 33, then
at absolute offers from these by proprieter values relation in staater to EE stack from a "f" is "deplik streaction" in "defentation" IS	nc keybind thody fullscreen toggle nc keybind thody prevdetile toggle	TE ATLAS DAY BEACHER -D DENIS -N"	<pre>Active the control of the contr</pre>	find -L.1 . /Wownto die/(N. S.). 5 (P. N.)'
stacks"s(stack_frame-1)*	A sone keybindings like 10 chang to different basks:	n atlas nhinguda Aethogs entin		
A same y y and	Mergistient -ii white read insection case Sine in	 A. A. A	<pre>Apr and Contains and ap PAPTH Dispectant sections are and contains the sections are and particular templation. Dispectant particular transmitter r =-esit K + Ds_ 1 esi (1.1) templation particular transmitter r =-esit K + Ds_ 1 esi (1.1) templation particular transmitter r =-esit K + Ds_ 1 esi (1.1) templation particular transmitter r =-esit K + Ds_ 1 esi (1.1) templation particular transmitter r =-esit K + Ds_ 1 esi (1.1) templation particular transmitter r =-esit K + Ds_ 1 esit (1.1) templation particular transmitter r =-esit K + Ds_ 1 esit (1.1) templation particular transmitter r =-esit K + Ds_ 1 esit (1.1) templation particular transmitter r == templation particul</pre>	10 ft
If terientation a "vertigel" in fatiger a "1" 11, then lacebrd, masterbru, energitetta "fartul", energiaster: "Startu"	<pre>case line to requer) to set sinces par (choice par-1) / trained there exit, not of sinces part (choice par-1) / trained there exit, not of sinces part (choice part) / trained (choice - part to the since part of the since part of the since part to the since part of the since part of the since part to the since part of the since part of the since part to the since part of the since part of the since part of the part of the since part of the since part of the since part of the part of the since part of the since part of the since part of the part of the since part of the since part of the since part of the since part of the part of the since part of the part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the since part of the s</pre>	(big() (as antigres "tilligres in gres,)	22 Genand Va PoonToggle accildoc 21 Grenap (C-B) MERDTreeToggle(CE)	The star ised "/*5/6"
If period the tempting and the states of the	set store for transformer and th t(rransforder_stdth-1) ([[-n 10 3] 64 [[-0 10 3] 64 santraj(13 -1 (2)cut -6' -43) 1	1 in the struct of the first structure of the structure o	an rai artis (note "permu" (b)/Cell -p "search (FPD.")
taciong, masterone, enptystachenstentigt", emptystasterentitentigt"	Pad 3 Shad 11 8 220 Dim pay	suce stattes "systax off" so it statter so	20 syntax teedlog 30 wextant fegelsestax() 31 ## eiter tegelse madie teeds 20 wereatien 21 wereatien 22 wereatien 23 wereatien 24 wereatien 25 wereatien 26 wereatien 27 wereatien 28 wereatien 29 wereatien 20 were	in betra fecte "parte" (bHDen or "search (Hobo") - case foot in eff t : t die t
trane a gatace 366 Integraceb 66 Anactopausterb ti (actopnasterb, Smactopstaceb,)	<pre>\$40*) \$40*\${\int_\$20*}; {\int_\$40*} {\int_\$20*}; {\int_\$20*} {\int_\$20*}; {\i</pre>	A R 101 BED ING WHITE INCIDE \$13.99	20 etsel Systax enable 1 endid 1 27 medianctian	The share a
	Perestclient chain \	Contraction of the second of the second of the second of	1 34 Addrew entr tagite 1 39 wettient Tagiteteree()	Can the state of t
And then gailed may then have () one name with the find current frame inth two prome. ()	set windte san Stan : set frane Border width 0 14	funde(\$(find -L - type d -mase "\$1" 10 /dee/milt)) ()	etse i set norightle/t keynapi 1 endir	starch 11
menstylneer in []] 00 m cycle -1 , shift -e sact , feest -e Sinact , cycle 1	wed 1 startes ast, hock maps is	founds (\$(findtype & -name "11" 25 /dev/rull)) If E -e "Sfound"], then	<pre>viction toulking (erg contact and surve) and (erg then within toulking (erg contact)) </pre>	The sharehidden by books and be been been by books and be been by books and be been by books and be been bee
MA GENE -2., SUD-T -4 LAC , FROM -4 DARC , SQUE 1 - DEBE FOR THE CONTROL OF THE SUD-THE START - TO SLARE YET - SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE - SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE SUD-THE	<pre>defg110(10(0)) + He - 1)) wrete(1) are case 1,</pre>	<pre>C (1) ***********************************</pre>	<pre>processing regions are being regions are being the list and the list of regions are being the list and angletter areas a cash of the list angletter areas of the list angletter areas wetting the being areas at any list between wetting the being areas of the list and the being areas of the list and the being areas and the being areas of the list and the being areas are the being areas of the list areas</pre>	al a browne fates not structury at 4 lise, the only way dog .
Stack = 0 1 th che faces - e sestern	n denet. N m anitze in altreetet tepnuts einerty	ei ofist	to et manications","	40 B = Drinke Files we stretchen at a lise the usin say but at a lists primarily meeter, when first is a city, pressil of a suprise section rests is result on an the preparation of a suprise section rests is result on an the preparation of a suprise section rest of the suprise press of the a suprise section rests withing this interact interact of a suprise of a suprise relief on the average of the a suprise of a suprise relief on the average of the suprise of the suprise relief on the suprise of the suprise of a suprise of the suprise relief on the suprise of the sup
			call Dualband; "a", "icall ToggleSuntas()(CE)	and most i contain and the ariting this waiterer interer. 21 m
Startyill" - "y" & 66 50 11 s part spart of me saler elienet maat, il Restyill" - "y" & 66 50 11 s part spart of me saler elienet maat, il " [-e "Startyill"], then 50, auto, size he focus -e salecht, 50, 11	 b. Stypic Statistics if	in collect) (<pre>interface (item: .testering .call fagine(stat())CD correct builded()',, .call fagine(stat())CD correct builded()',, .call fagine(stat())CD correct builded()',, .call setting item call builded()',, .call setting cent call builded()',,,, .call setting,,,,,,,</pre>	6: Breast () C = "Sact"); 6:
[-2 "SEmpty211"], here 20, were, rise he focus on subsection, s.B. 11 	hc keybind Stor-semicates spawn Sconf/twap auto hc keybind Stor-stack spawn Sconf/twap close	<pre>if { { { { { { { { { { { { { { { { { { {</pre>	the stat "call building " ing.", " set" setting."(cr>')" + endler	
Contemplates	4510510799.91	read -F(1) (Franker(51)) read -F solitist of "Sfranker(51)" fill solitist of "Sfranker(51)" fill read -Fill solitist of "Sfranker"	14 For the first for the second se	and the set in set
her , feres -e fastler? , thift -e falacit , facus -e fastlert, till	(A)LCONTPERT Regions Standantrat-b realise Lord -SHISIZCOTTP bc regions Standantrat-b realise down -SHISIZCOTTP bc regions Standantrat-j resize down -SHISIZCOTTP bc regions Standartat-realise up -SHISIZCOTTP c regions Standartat-realise regions -SHISIZCOTTP bc regions Standartat-realise regions -SHISIZCOTTP (Shi regions Standartat-realise regions -SHISIZCOTTP)	17 Buillie of "Stream [Sec1])"	to for E ter, 41 3 in Effat, fasciliner3, Efer, fagtheart, Efer, fan ene Therman Cleaderon, teg - ist filetgreet at rocto tendfor	eist 0 Time
	he kegoine shoe-control-1 resize right -stesseetre	is /	the far key is ("a", "a", "a", "a", "a", "a", "a", "a	
D. C. WIR closer climit. 4111 serier of ready, craves start of two start properties and the start of the s	by septembind	a i resets lig siat of the framebuller for the lig		FLAC FLAG
aphysicar + -)] to ace , recut + prizets , tabr + parties and	to sourcebindall The sourcebind phorbattal sour The sourcebind phorbattal sour The sourcebind phorbattal relian	C -s	1.55 mot up pow output (now year) (prevent) 51 moterus (cluster) in , toh(CD 1.55 vooreus) (cluster) in devo(Cor) in , toh(CD) deg "v#jed "v 1.56 moterus (cluster) im , thythex(CD) 1.57 vooreus (cluster) im , thythex(CD)	122 % drie practicy 12 the default heliavker, but 5 also have 2 643 % allocated to "pres", so when invested that was, 11 mem or
tt a ne sawrining ynd ant a	a e faus	22 N DEPEND & DEEL SECTION WITH UN DELINARY COMMAND	TE WAREAU (LEADER) C "TY	2.25 w Line controls 12.5 the Derivation Instruction, but C. Albo Auver C. Control and C. C. Santa and Controls C. Mail and C. C. Thomas Dr. Control and C. Santa and Controls C. Mail and C. C. Thomas Dr. Control and C. Santa and C. Santa and C. Santa and C. Santa and C. Santa C. Santa and C. Santa C. Santa and C. Santa C. Santa and C. San
	+ hc keybind shot-Tab cucle_all +1	a ta tacka té é sand prins ter jarah pilitinaries	To veloceasy (Leaderic "vy To Severasy (Leaderic "vy To Call Guilland, "Pr. "vy") To Call Guilland, "Pr. "vy") To Severasy (Leaderie "vy	
	the Registed Storve cycle	en m hanka më a sand unive len levul dillinaries 27 MB() (dill'99"(vikpeger, sleer,) 28 MB Lores ve a eret oster races alcovering m der)((dill-edillo)(or 95")eskajate, dikar,))	The delivenceson dualband	in exit
81 1 1 1 1	he keybind short focus up he keybind short focus up	the state of the s		
12	he keybind shot-1 jumpte urgent he keybind shot-Shift-h shift teft			
	The traject sub-sub-field spin-sub- tion of the sub-sub-field spin-sub-sub-sub-sub-sub-sub-sub-sub-sub-sub			
		constituental faces at the att		1.1 ··· · · · · · · · · · · · · · · · ·

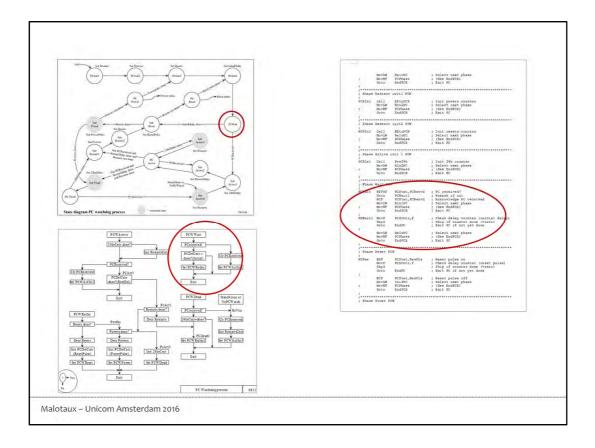
Can you really see structure in code?

If you work only in code, producing bugs is to be expected. Bugs shouldn't be there. Hence, don't work only in code.



If we don't have a proper design, the developer should still have a model of the system in his mind. A problem is that the model in the mind is fading, so we have to continuously refresh it. That's why you see developers moving and switching screens all the time, trying to keep the model in their mind current. At the same time they should in their mind correlate the model in their mind with the code they should produce or modify,.

If they would get the model of the system out of their mind e.g. on paper, or whiteboard, they don't have to continuously refresh it, they quickly see flaws in the design, others can even watch it and see possible issues, and they have their whole mind available to do the thinking and correlating.



Simple example of correlation between a state diagram, the derived flows, and the resulting code. Can you see what's happening in the code? I can't.

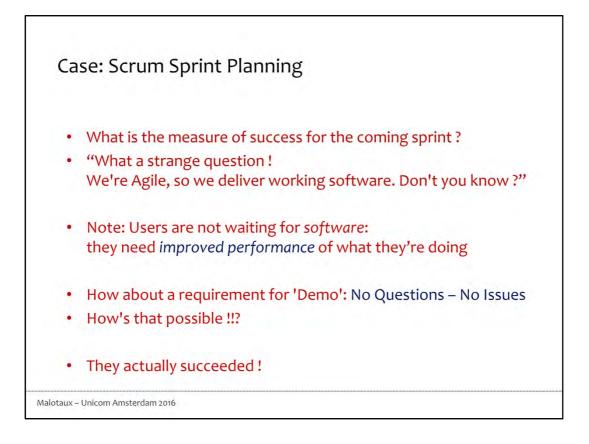
Can you see what's happening in the flow diagram? A bit better.

Can you see coherence in the state diagram? I can. And I easily can reason with this with anyone, even the CEO.

It took me quite some time to compose the state diagram. But then the flows and the code were don quite straight-forward and had no errors whatsoever.

The product worked for years without any problem.

It also adheres the the requirement: "After a year I should be able to add or change anything in this system without introducing any issues, being up and running within a day".



I came in an project of some 70 people, with 3 Scrum teams of some 12 people each. We know 12 is too many, but that's another story.

At a Sprint Planning meeting I asked one of the teams: "What would be the measure of success for this Sprint?"

They looked at me: "What a strange question. We're Agile, so we deliver working software. Don't you know?"

I asked: "Shouldn't we have a measure of success, to know that we really did a good job?" and suggested: "No questions, No Issues". That's easy to measure: one question or one issue and we know we failed. No question and no issue and we know we were successful.

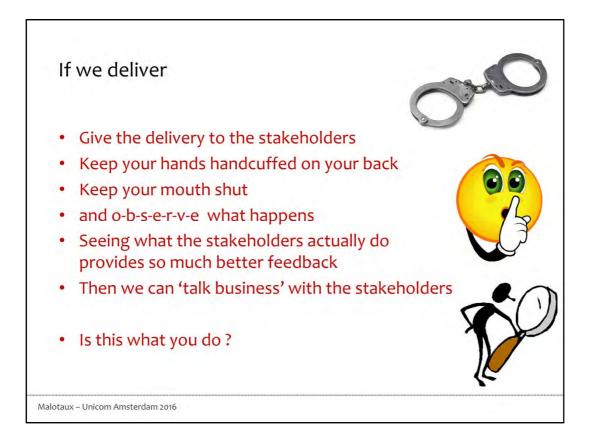
Their first reaction was: That's impossible! Surely there will be some questions when we deliver and there are always some issues.

I suggested: "You find out how to do it. It's just a simple requirement: "No questions, No Issues".

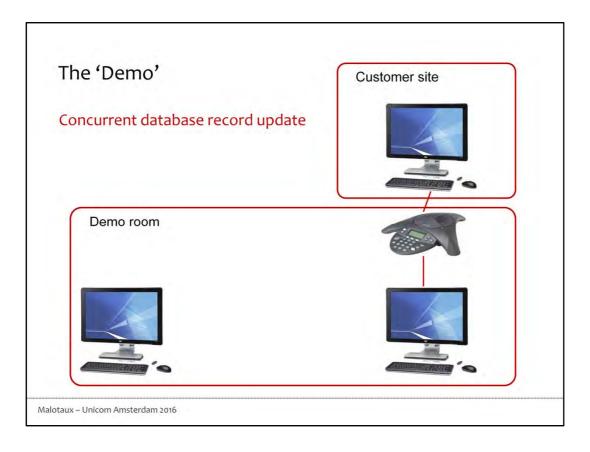
Interestingly, they immediately started thinking how they could deliver according to this requirement.

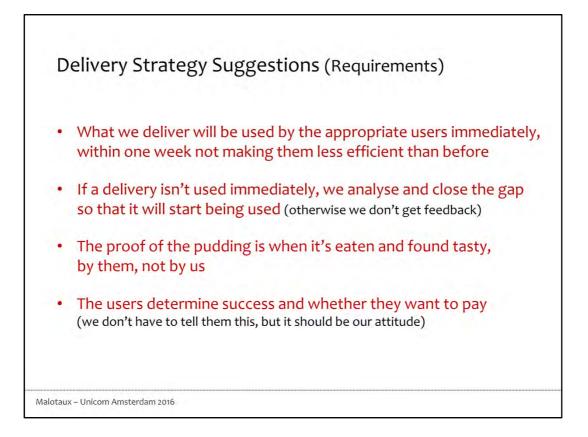
For example, someone thought: "Ah. Perhaps halfway the Sprint we ask someone to check it out and to see whether he would have any questions?" I said: "You're on the right track. Just find out how to do it. The requirement is simple."

Actually, I didn't expect them to be successful in this first Sprint, perhaps after a few. Surprisingly, they were successful.

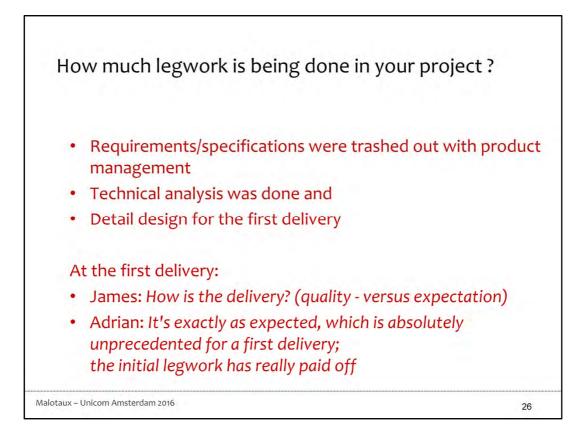


Scrum reviews usually only demo. How about using the advice as shown here. Is this what you do? If not, why not?





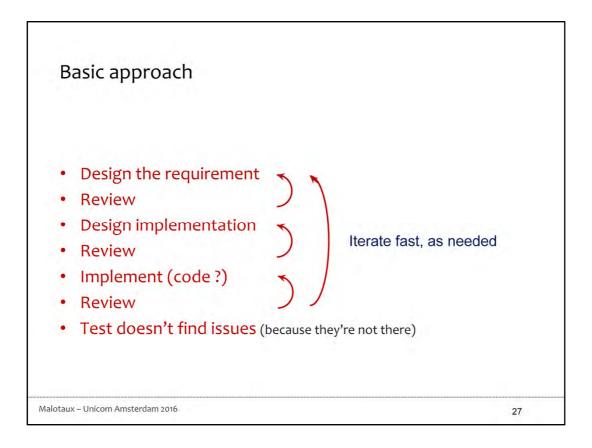
This is my suggestion I gave the architect and the project manager to consider as Delivery Strategy.



This case was an organization with extraordinary bright people. In many projects we have to explain things over and over again, but in this project people needed only half a word to understand and do things better.

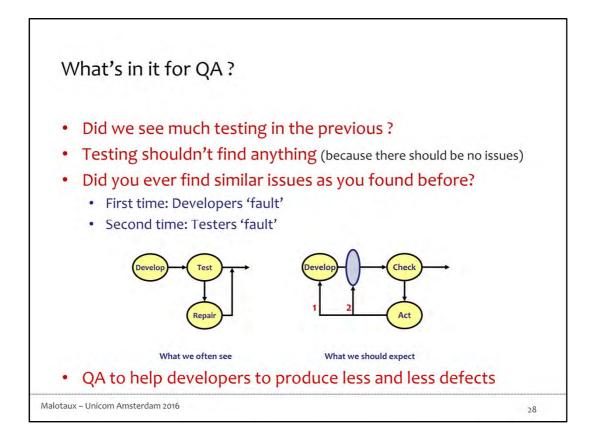
James (their new QA person) told me this story. He asked them to prepare well, design properly, and then do the coding. The result: "It's exactly as expected, which is absolutely unprecedented for a first delivery."

He suggested it, they did it, and it worked. It's great for a QA person to work in such a fertile environment!



The basic approach.

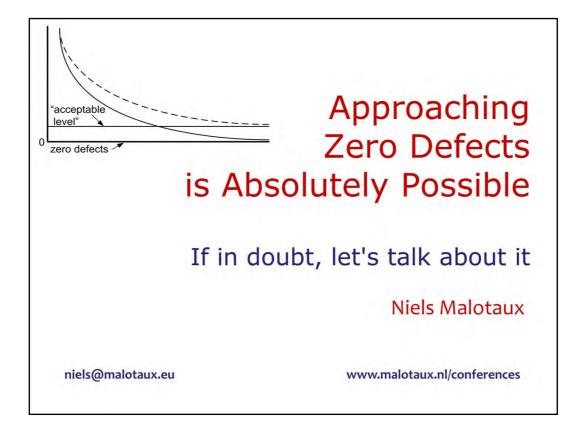
Some people simply call this: 'engineering'.



So, what's in it for QA?



What will you do next?



If in doubt, let's discuss. To you it may be theory. To me it's reality.