

Gaia ECSS in the eXtreme

update of 2008 talk at ESOC

William O'Mullane

LSST
Tucson
Arizona

Jan 11th 2017 DMLT Tucson



- Herein lies the crux of science processing problems for me
- In a meeting in 2007 I was asked :

What do you think you are doing, a science project or something ?
Well yes I rather think I am!

did not seem to be the expected answer.



- Herein lies the crux of science processing problems for me
- In a meeting in 2007 I was asked :

What do you think you are doing, a science project or something ?
Well yes I rather think I am!

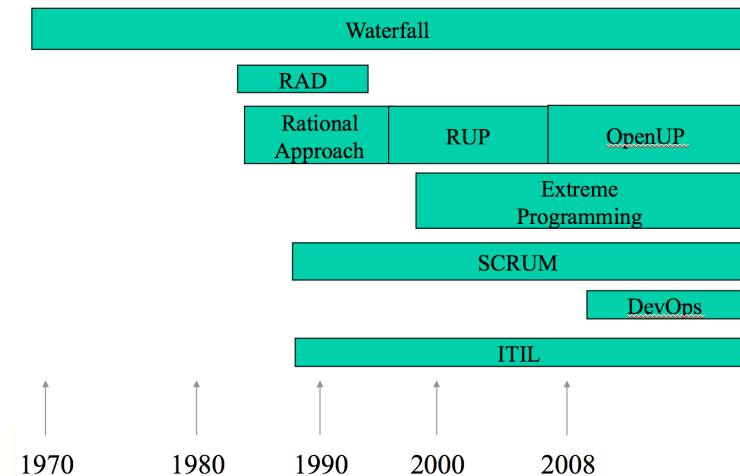
did not seem to be the expected answer.



- To clarify In the ESA context think different to satellite development
- i.e. the waterfall
 - Requirements
 - Design
 - Implement
 - Deliver
- Precisely in that order and precisely once
- Genuinely how many of you ever do this ?
- Not clear any large software is done in this manner anymore

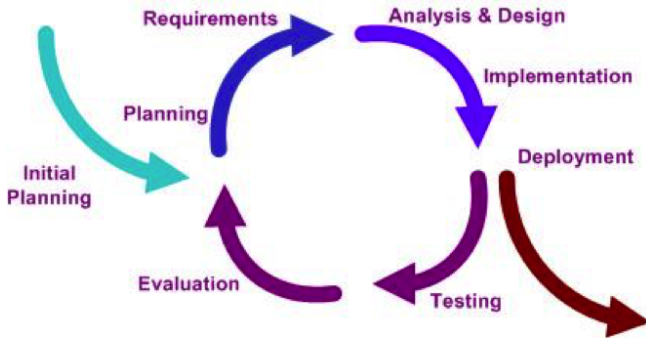


- OO Development Methodology Timeline



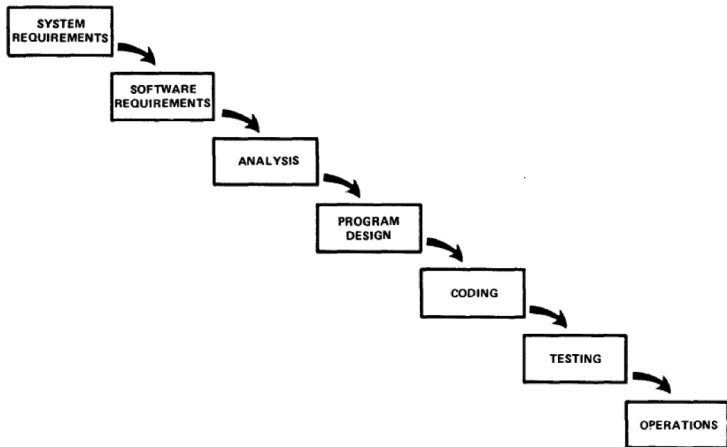


- Problems with Traditional Waterfall
 - Customer doesn't see results for a long time
 - Sometimes delivery is not what was wanted
 - Often the requirements have changed
 - Especially when the customer sees the implementation
- Enter Iterative methodologies



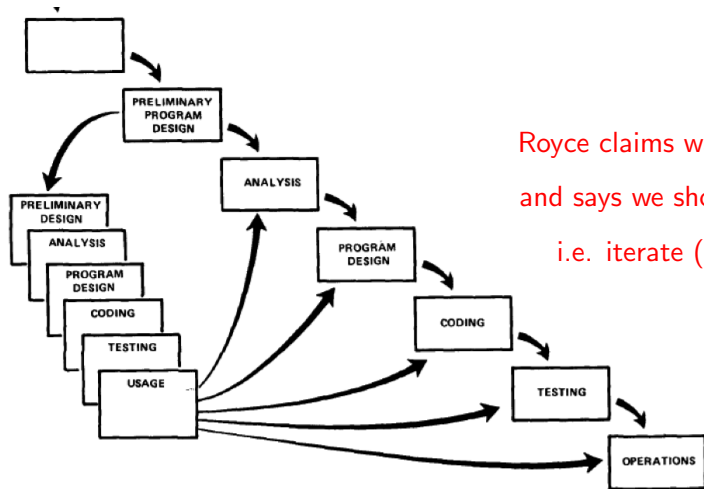


- Royce (Royce, 1970).
- Considered inventor of term waterfall (his fig)





- for the managers that actually read the next page of the paper



Royce claims waterfall is flawed
and says we should do it twice!
i.e. iterate (his other fig)



- Do things faster and cheaper with nimble teams
 - eXtreme Programming (more to come)
 - SCRUM huddle and watch the backlog
 - CrystalClear Cockburnss undisciplined XP
 - several others
- Rapid Application Development (RAD) does not cut it in Agile
 - James Martin seen as the father of Agile
 - but he did not go to Utah



- One of the so called Agile techniques
- This is not how it is .



Copyright © 2003 United Feature Syndicate, Inc.



- Some say nothing new started in 1950:
 - Larman, C., Basili, V.R., 2003, IEEE Computer, <http://www2.umassd.edu/SWPI/xp/articles/r6047.pdf>
- eXtreme Programming or XP was created by Kent Beck during his work on the Chrysler Comprehensive Compensation System (C3) payroll project around 1997
- in October 1999, Extreme Programming Explained was published
 - Several other books followed
 - And many tools (which you already use)



- Principally reduce the cost of change
 - Actually keep cost fixed while adapting the system
 - Embrace change !!
- Communication
 - light documentation & metrics
 - Simplicity - KISS principle
 - Feedback
 - From the code via unit tests
 - From the customer (product owner - scientist) via acceptance tests
 - From the team by meetings
- Courage
 - Dont be afraid to Refactor (The tests will save you)
- Respect
 - Team ownership of code
 - Shared responsibility we all succeed or we all fail!



- We plan once a month using points
- Yes we wrote stories on post-its they are costed in points
 - (1 point \approx half a day)
- Each team member uses their points to buy stories to work on for the month
- Allow less experienced people perform a task - they may have to be allowed more points
- the story at LSST (?).



Copyright © 2003 United Feature Syndicate, Inc.



- Costing is done as a group
- All stories were put in the XP tracker twiki where everyone *recorded the time they actually spent* on the story
- We get better at knowing how long something actually takes.
- We sometimes sit at one computer but usually when there is a problem not as a norm
- We often sit with a projector and review code in a group we could do that even more often



Infrastructurereiteration < Tracking < TWiki - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://gaia10.8000/twiki/bin/view/Tracking/Infrastructurereiteration

java api Zoom Astronomy Glos... weather.com - Local ... GaiaAGIS < Tracking ...

sun coding java static v... Talk:CU1:Coding standa... Yahoo! Mail - womullan... Infrastructurereiteration ... Ecologia Certificada - Pro...

Infrastructure for properties, extended to SQL statements stored outside of application code	1	1	0	😊	100%	0%	Complete	JohnHoar
SpecialUtilitiesStory Design and implements a couple of special utility classes	2	2	0	😊	100%	0%	Complete	UweLammers
UpdateArchitectureStory Update the Architecture Document	4	2.5	0	😊	100%	-37%	Complete	JohnHoar UweLammers WilliamOMullane
WhiteBoardObserverStory Monitor contents of Whiteboard and schedule processing tasks for execution	3	3	0	😊	100%	0%	Complete	UweLammers
Team totals	26.4	21.7	0	😊	100%	-17%		

Create new story in this iteration

Developer velocity for iteration Infrastructurereiteration

Who	Assigned	Spent	Remaining	Completion	Assigned Tasks	Remaining Tasks
WilliamOMullane	11	9	0	100%	12	0
UweLammers	11	8.5	0	100%	13	0
JohnHoar	4.4	4.2	0	100%	7	0

Done



We have a long term plan to launch with more firm stories for the six month cycle we are in

- Work is planned in detail monthly
- We concentrate on what we need this month
- We know quickly when something is much *tougher* than expected
- We do not design for what we might want next year we will do that next year
- This keeps the code base at a minimum
 - Also most code is in use !
- This has been essential to keep system performance at peak
 - Immediately obvious if new features slow us down.
- Retrospective feedback
 - meeting had to be included as non discounted points ..
 - meeting preparation got included
 - science projects were included - a work package was made for that but not reported



Why plan?

- In a complex development enables tracking of planned schedule and better management of critical path WPs.
- Planning:
 - Ensures we work on what needs to be done (even the bits you dont fancy working on!)
 - Focuses individual effort
 - Sense of accomplishment and motivation
 - Fine tuning and prioritisation of work

Why track?

- Need to measure how were doing against the plan somehow
- Tracking:
 - Monitor effort spend on WPs (defined in Cost at Completion)
 - Uncover potentially incorrect/unrealistic estimates
 - Feedback to improve planning/estimation skills
 - Progress reporting at the WP level



- Define period start and end dates for iteration (no points for planning meeting day) - Set up Google sheet
- Estimated column completed by **controller with team members at pre-planning meeting**
 - Points based planning per Story (WP), 1 point 0.5 days
 - Points available are personalised by taking into account:
 - Leave days removed
 - 20% - 30% discount in general since no one is 100% productive
 - Greater discount for top management but no one can opt out
 - Meetings, training/conference etc. fully tracked but NOT discounted
- Spent column updated by team by end of iteration
 - Unplanned stories can be added
- Retrospective with whole team to look at previous & next iteration
- Convert points spent per story into man-month per WP
 - Account for any sick days/unplanned leave
- Transfer over to monthly tracking sheet generated from CaC (Cost-at-Completion) Manpower Planning (use special formula)
 - Enables tracking of spend per WP over a period of one year



- This type of approach only works with lots of test code and continuous integration
- That means automatic tests like Junit
 - CruiseControl builds/tests a few times a day
 - emails everyone if the build or tests fail
- Trying to get ≥ 80
- AGIS just for interest
 - 100K lines of code
 - 30K lines of test code on top of that
 - 95K lines of comments



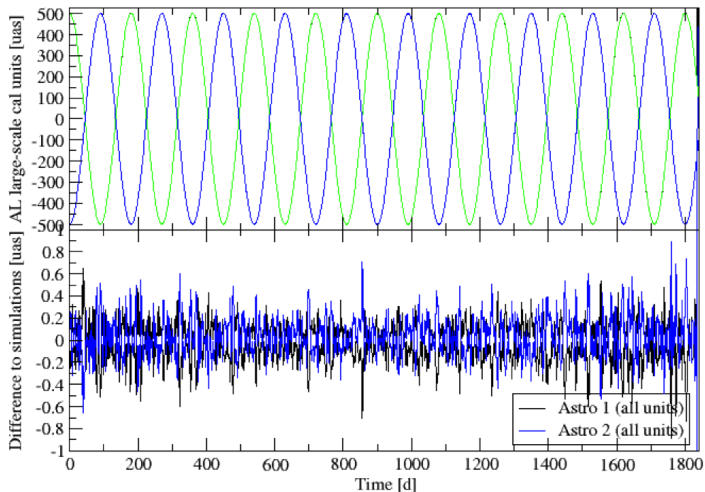
Traditional contract with first delivery after 2 years still failed to produce working system after 4 years

ESAC team produced first functioning AGIS within 4 months with 5 iterations

- Setup - 2 weeks machines/reviewing/looking
- Infrastructure 2 weeks, DBMS/data access
 - Project Scientist visit (our product owner)
 - Algo 3 weeks, first algorithms
 - Scientist expert visit/ pair programmed
 - Integration 3 weeks -Underestimate 37
 - MakeltWork 3 weeks - refocus after missed goal
 - Test/Verification report published



Simulated long term basic angle variation recovered with μas accuracy



Plot from Uwe Lammers



European Cooperation for Space Standardization

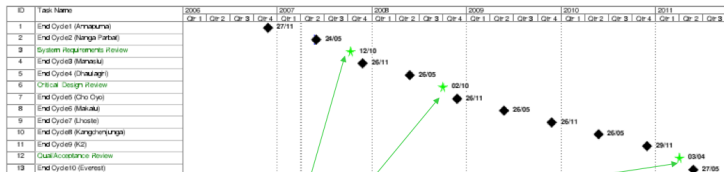
- Tells us how to do things and what documentation to produce: many books
 - ECSS-E-10B Requirements
 - ECSS-M-30A Phasing
 - ECSS-M-40B Configuration Management
 - ECSS-Q-20B Quality Assurance
 - . The list is long, very long
 - NOT unlike ITIL in this sense ..
- Also reviews waterfall style !
 - Not surprising this is mainly concerned with satellite construction.



- ECSS actually very flexible
 - Like that SRS etc. allow mix design and requirements great for infrastructure
- But that means care must be taken
- Standard must be tailored for the project.
- Which documents will you produce when etc
- Does this exist for LSST ?? - you have some if it in LDM-493



- At least 3 or 4 before launch typically
 - System Requirements Review
 - Critical Design Review
 - Qualification Review
 - Acceptance Review
- Within DPAC there are six month development cycles more XP style
 - Docs updated in each cycle concentration before review on docs for that review.
 - Latest released version of documents just pulled form the livelink and handed over - *no particular extra work*

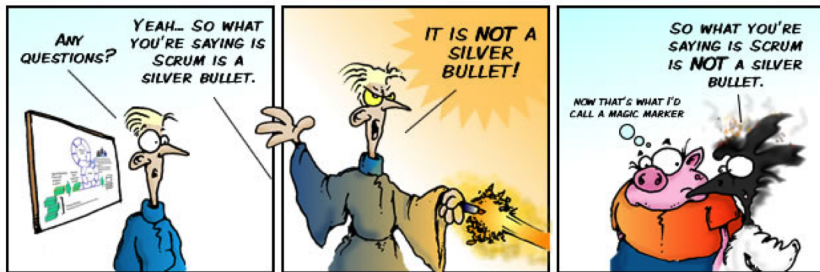


ECSS Reviews



- Encouraging experiences from AGIS
- Team feedback suggests leadership important
- XP is more process oriented than "normal" development
- So managed to pass ESA reviews for the entire DPAC system
- We have tried to pick useful documents from ECSS
- We do need documentation it is nice to have a set of templates for these
- Still pressure for traditional planning and accounting - care to be taken on this one
- For sanity one should read Dilbert every day

Dilbert cartoons are produced by Scott Adams see dilbert.com



By Clark & Vizdos

© 2006 implementingscrum.com

Remember nothing is perfect



① Acronyms

② References



The following table has been generated from the on-line Gaia acronym list:

Acronym	Description
AGIS	Astrometric Global Iterative Solution
DPAC	Data Processing and Analysis Consortium
ECSS	European Cooperation for Space Standardisation
ESA	European Space Agency
ESAC	European Space Astronomy Centre (VilSpa)
ESOC	European Space Operations Centre (ESA)
IEEE	Institute of Electrical and Electronic Engineers
LSST	Large-aperture Synoptic Survey Telescope
OO	Object Oriented
RUP	Rational Unified Process
SRS	Software Requirements Specification
TOC	Table of Contents
VR	Radial Velocity
XP	eXtreme Programming



① Acronyms

② References



Gill, R., Gracia, G., Lupton, R.H., O'Mullane, W., 2014, In: Modeling, Systems Engineering, and Project Management for Astronomy VI, vol. 9150 of SPIE, 91501E, ADS Link

Royce, W., 1970, In: Proceedings of IEEE WESCON, 1-9,
<http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf>