



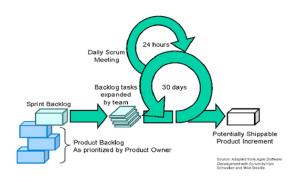
### Estimate with Confidence<sup>™</sup>



**Agile Estimation for Space Software** 

## Agenda

- Introduction
- Agile Software Development
- Agile in Space
- Size and Cost Estimation for Agile Software
   Development
- Wrap Up



#### **Agile and Scrum Process**



## Introduction

- Agile development practices have enabled organizations to deliver quality software that optimizes customer satisfaction
- But is agile for every type of project
- Space and other mission critical software have high reliability, fault tolerance requirements with strict safety and performance criteria
- Organizations developing space based software are looking for ways to do development faster, better and cheaper
- Can agile development practices facilitate this requirement





## Introduction

- Back in the day ... Complexity of applications was overshadowed by the logistics of implementation
- Technology improved ... today software solves increasingly complexity problems
- The so called 'software crises' (mid 60's to 80's) resulted in many 'silver bullet' type solutions
- Lots of smart software development professionals began looking for more lightweight methods to address complexity in achievable chunks



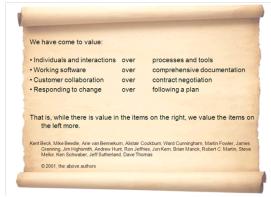




## Agile Software Development

## **Agile Manifesto**

- We are discovering better ways of developing software by doing it and helping others do it
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
- All agile projects adhere to this manifest
- All agile projects share a common set of principles
- Each agile project uses a unique set of agile practices to implement these principles
- Successful estimation for an agile project is like software estimation for any project – you need to understand the project properties and the practices employed





### **Traditional Software Development**

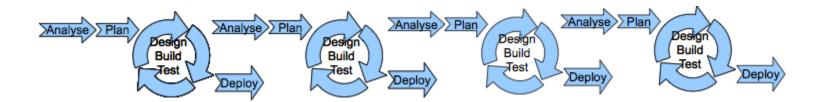
- Requirements are analyzed
- Architecture and design are created
- Requirements are implemented, tested and delivered
- Months (or longer) occur before there is usable software for the customer to evaluate





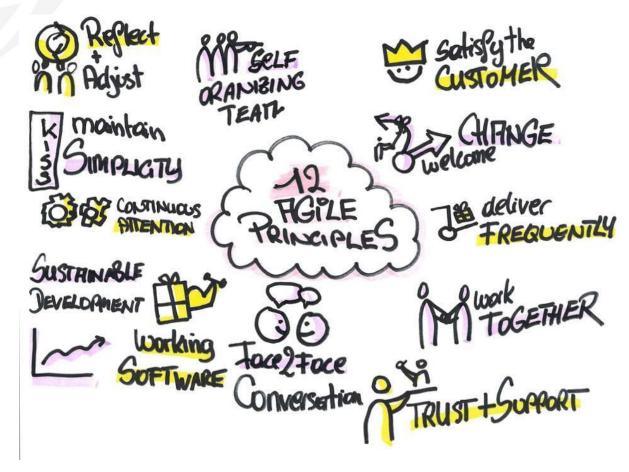
## **Agile Software Development**

- // Usable chunks of software are developed in short periods of time (sprints, iterations, etc.)
- Requirements are translated into user stories and become the project backlog
- User stories deliver business value and are small enough to complete in an iteration
- Customer works with team and reviews software regularly
- Each iteration focuses on the user stories that are currently the highest priority of the customer
- Priorities may shift from iteration to iteration
- Agile teams expect and embrace change





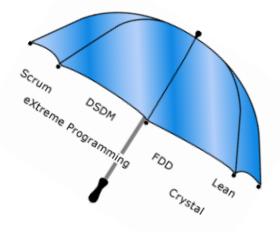
### **12 Guiding Principles for Agile Development**





## **Common Agile Practices**

- Pair programming
- Continuous integration with automated testing
- Test driven development
- Daily stand up meetings
- Co-located teams
- Code refactoring
- Small releases
- Customer on team
- Simple design







## **Agile in Space**

# Agile in Space

#### "NASA was agile before agile was a common term"

- Jim Highsmith one of the 17 original authors of the manifesto worked for NASA at one point
- In 1962 John Paup was a senior NASA manager planning

#### part of the Apollo program

• First thing every morning all key people reported to his office for a stand up meeting

### NASA Ames – Mission Control Technologies

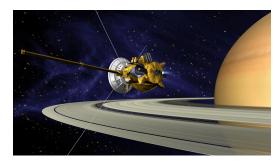
• Adopted a hybrid agile solution – segregating activities constrained by mission criticality from those more standard development activities

### Cassini Mission

• 2015 (more than 10 years after the mission started) – the maintenance team has adopted a hybrid agile process for software changes

### • Software Probe Plus – built by JHU/APL

• Several of the ground system software module teams are adopting agile practices





## **Challenges to Being Agile in Space**

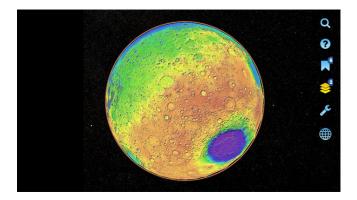
- Requirements for Compliance to industry standards and sponsor requirements
  - CMMI
  - AS9100
  - NASA Software Engineering Requirements (NPR 7150.2B)
  - European Cooperation for Space Standardization (ECSS)



- Requirements for detailed documentation
- Requirements flexibility (or lack there of)
- Detailed up front planning
- Requirements for specialized capability (as opposed to agile teams composed of generalists
- Formalized customer interfaces

# But agile is a philosophy not a development process

- Hybrid applications make the most sense for space systems
- Agile practices that make sense
  - Small teams evolving product in small visible steps
  - Daily stand up meetings
  - Pair programming
  - Continuous automated testing
  - Test driven development
  - Collaborative planning (including the customer)
  - Agile practices less likely to make sense
    - Evolving requirements
    - No formal up front planning
    - Little to no documentation
    - Refactoring







## Agile Cost Estimation

## **Agile estimation**

- Frequently asked questions
- How to estimate size for an agile project when the team is working with Story Points?
- What other cost drivers are indicated for an agile development project?





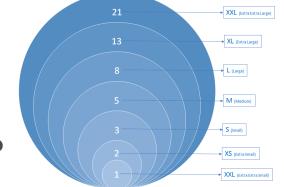
# **Agile Size Estimation**

#### Agile teams do a lot of their own estimation

• High level estimation as the backlog is created in the beginning of a project (Sprint 0)

## Estimates are notional and only make sense to the team

- Story points
- T-Shirt size
- Estimators challenge is to translate the teams knowledge into a size measurement that relates to their Cost Estimating Relationships (CERs)
- In the context of a parametric model agile size measures actually combine two typical cost drivers
  - Size
  - Complexity



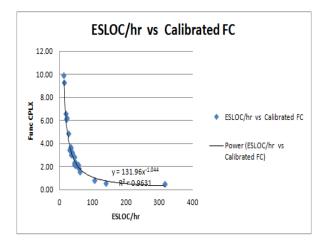


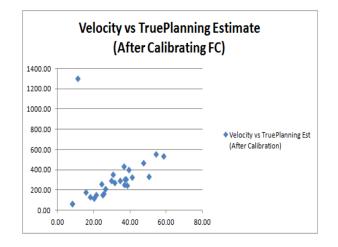
### Fortunately agile teams collect lots of metrics

12		- 21	-	2010 Agile Me	etrics_NEW_v3 (2	2).xlsm - Micro					Tools												
Home	Insert Pa	ge Layout	Formula	as Data	Review V		Loper -	Add-Ins X	Acrobat	t Des J(	ign Lave	JC	at									(	0 - 1
🔏 Cut		ori (Body)			= = >	- Wrap		General		-			vormal 2	Normal	В	ad	Goo	d		- 1x	Σ Auto		a i
e Copy	в	ΙŪ-		• A • 💻		Merge	& Center		%,	Q0. 0.	Conditional	Format	Neutral		tion	heck Cell	Exp	lanatory	Inse	rt Delete F	ormat Fill	Sort &	
I Format	Painter						or center			.000	Formatting *					incen een			-	*	👻 📿 Clea	Filter * S	
Clipboard	Gi .	Fo	int	G	Aligi	nment		≊][ N	umber					St	/les					Cells		Editing	
ecurity Warnii	ing Macrosha	ve been disa	abled.	Options																			
ch unt d	-	2																					
Chart 8	• ( <u>°</u>	f <sub>x</sub> C	D	F	F	G	Н		J	K L	M	N	0 P	Q	R	S	т	Y	7	AA	AB	AC	-
A	D	U	U	E	F	G	п		J	r L	Team	IN	UP	Q	ĸ	3		T	2	AM	AD	AC	*
											Status /												
Begin	End Iteration I	teration	Acceptan		Unit Tanta	Burn Up	FTE I	Pts /FTE		BA+ D CR	Full Time	Hours Recorded	Hr / Pt	Base	New	Modified	Deleted	ESLOC	ESLOC / Point	ESLOC / Hr	Calibrated FC	Calculated FC	т
Iteration 1/29/2010	2/11/2010	ateration 3	Tests 244969	Velocit 47.50	y Unit rests 1395	123.8	18.0	2.64	7.5	5.5 5.		415.50	8.75	842,310	10,440	1,650	768	12,133	255	29	3.45	3.92	-
2/12/2010	2/25/2010	4	248055		1369	178.3	16.0	3.41		5.5 4.	-	563.00	10.33	852.035	4,750	2,012	1,495	7,205	132	13	9.29	9.22	
2/26/2010	3/11/2010	5	251830		1369	194.0	13.0	1.21		5.5 1.	-	174.50	11.08	855,235	4,927	1,753	2,555	7,943	504	46	2.36	2.45	
	3/25/2010	6	256430		1359	220.5	13.0	2.04		5.5 1.		230.00	8.68	857,711	3,486	3,364	2,522	7,606	287	33	3.68	3.42	
3/26/2010	4/8/2010	7	260189		1353	251.3	12.0	2.56		4.5 1.		367.00		858,657	2,812	1,084	1,458	4,591	149	13	9.90	9.44	
4/9/2010	4/22/2010	8	257980		1303	292.5	12.0	3.44		4.5 1.		338.75		859,842	4,504	864	3,516	7,570	184	22	6.24	5.15	
4/23/2010	5/6/2010	9	258110	31.50	1316	324.0	12.0	2.63	6.5	4.5 1.	960.0	261.25	8.29	860,999	6,153	617	2,448	8,299	263	32	3.38	3.57	
5/7/2010	5/20/2010	10	264997	24.50	1313	348.5	12.0	2.04	6.5	4.5 1.	960.0	236.50	9.65	864,704	7,533	3,065	1,119	10,462	427	44	2.31	2.52	
5/21/2010	6/3/2010	11	282924	25.50	1313	374.0	12.0	2.13	6.5	4.5 1.	960.0	160.50	6.29	870,645	5,048	3,771	2,129	9,178	360	57	1.86	1.93	
6/4/2010	6/17/2010	12	288722	21.50	1316	395.5	12.0	1.79	6.5	4.5 1.	960.0	165.00	7.67	873,564	2,650	2,454	4,153	7,275	338	44	2.84	2.53	
5/18/2010	7/1/2010	13	288397	37.75	1316	433.3	12.0	3.15		4.5 1.	960.0	278.75	7.38	872,061	9,832	2,372	2,289	13,095	347	47	2.11	2.37	
7/2/2010	7/15/2010	14	288081	34.50	1314	467.8	12.0	2.88		4.5 1.		311.50	9.03	879,604	3,268	1,318	2,334	5,824	169	19	6.61	6.20	
	7/29/2010	15	209407	18.00	1313	485.8		1.29	7.5	5.5 1.	1120.0	7	7.61	880,538	2,371	1,862	1,240	4,542	252	33	3.59	3.41	
7/30/2010	8/12/2010	16	2107	Velocity vs ESLOC								119.		Velo	city vs /	Actual	Dev H	lours		7	2.41	2.39	
8/13/2010	8/26/2010	17	2158				,	0200				132.			city voi	Actual		iours		15	0.83	1.02	
8/27/2010	9/9/2010	18	3119	30,000								269.	600.00			•	_			5	4.88	4.40	
	9/23/2010	19	3042				•					344.			817x - 20.87	Ľ,				b	3.31	3.22	
	10/7/2010	20	3046 3058	25,000	•							524. 77.	500.00	R	=0.8516	/				6	2.15 0.50	2.22 0.32	
	10/21/2010 11/4/2010	21 22	3063	20,000				_				318.	<b>≦</b> 400.00		• •					6	2.18	1.96	
	11/18/2010	22	3067		•							272.	P 400.00		• • / ·					7	3.24	3.01	
	12/2/2010	23	3085	15,000		0000 0 00133						52.	300.00		• • / • •			<ul> <li>Velocity</li> </ul>	vs Actual Ho	urs 9	0.57	0.76	
	12/16/2010	25	3104	80		259.2e <sup>0.0122x</sup>	-			city vs ESL		283.								4	3.06	3.33	
	12/30/2010	26	3109	10,000					Expon. (Velocity vs ESLOC)			60.	¥ 200.00		,			Linear (Velocity		tual 1	1.56	1.80	
otal			310,99									5,889.						Hours)					
verage			272,30	5,000	• • •	, T						265.	100.00				_			95	3.49	3.42	
				0										ø.									
				0.00	20.00	40.00	50.00	80.00					0.00	00 20.00	40.00	60.00 8	30.00						
ev R	6			0.00		Velocity				Cha	t Area		0.0	20.00	Velocity	00.00 0	0.00						
E	4	2													renouny								
oc	1.2	10																					
ev Man	0.3	20				E	SLOC/	hr vs	Calik	orated	IFC												
R Man E Man	0.1	20 20			12.00					_													
oc Man	0.2	20																					
					10.00	:				-													
► H / Unit	Tests / Ptsp	erFte / H	rsperPt	Raw Data	Cost Objects	Activition	Decourse	Poor	accion	Drotoby	e / 知 /			14									1

## **Agile Size Estimation**

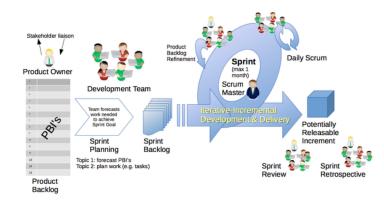
- Study of PRICE's agile data found no correlation between story points and software size or effort
- Did fine a significant relationship between software size and complexity (Functional Complexity in the PRICE model) pairs and effort.







- The fact that your project is agile is not a cost driver
- There are potential cost implications to adopting agile practice
- Estimation team needs to determine which agile practices apply





- Agile teams tend to be highly skilled
- Hard to be a slacker in an agile environment
- Working closely with high skilled team members, learning curve for new members is quick
- Input parameters to your model indicating team experience would be affected
- Agile teams tend to have tool sets that are quite sophisticated
- This would be especially true on teams working with space systems as it would greatly facilitate compliance to standards
- Input parameters around tools or automation would be affected

### 🔷 Jira Software



GitHub

- Co-location of teams should improve team productivity
- Culture of interruption
- Questions answered in real time
- Team cohesion increases
- Co-locating stakeholders and SMEs with development team creates a real time IPT
- Well run stand-up meetings increase productivity and quality
- Cost drivers indicating distribution of team and communication practices would be
   affected







- Continuous integration with automated testing should increase delivery productivity
- Important in space systems to maintain safety critical compliance requirements.
- Code is checked in frequently and builds are run and test regularly before developers forget what they changed
- Red tests raise red flags team fixes them right away
- Since little code is changed, errors are east to track down
- Fixes occur quickly
- Cost drivers focused on integration test complexity would be affected

## Conclusion

- While not all agile practices make sense for space systems development, there are many that can (and have) improved the ability to deliver high quality space system software
- A hybrid version of agile is most appropriate for safety critical software
- NASA has been successfully employing agile on many programs for many years
- Estimating an agile program is no different that estimating any other software
- Understand the program and the process being employed
- Study data from previous similar programs
- Discuss project particulars with the delivery team



