Overview of Cleanroom Software Engineering

Paul L. Jones, CDP, CSQE
FDA/CDRH/OST
V: (301) 443-2536 x164
Email: pxj@cdrh.fda.gov

What is Cleanroom Software Engineering?

Why the name - “Cleanroom”?

Cleanroom SE History

Late 1970's thru Early 80's	Theory developed
Mid 80's -	Theory applied
	ARPA STARS Program selects Cleanroom SE
1990 -	IBM Cleanroom Software Technology Center established

Cleanroom SE History cont'd

1995	Operations research theory applied to Usage Model
1996	Enumeration theory developed
1996	Cleanroom SE mapped to CMM

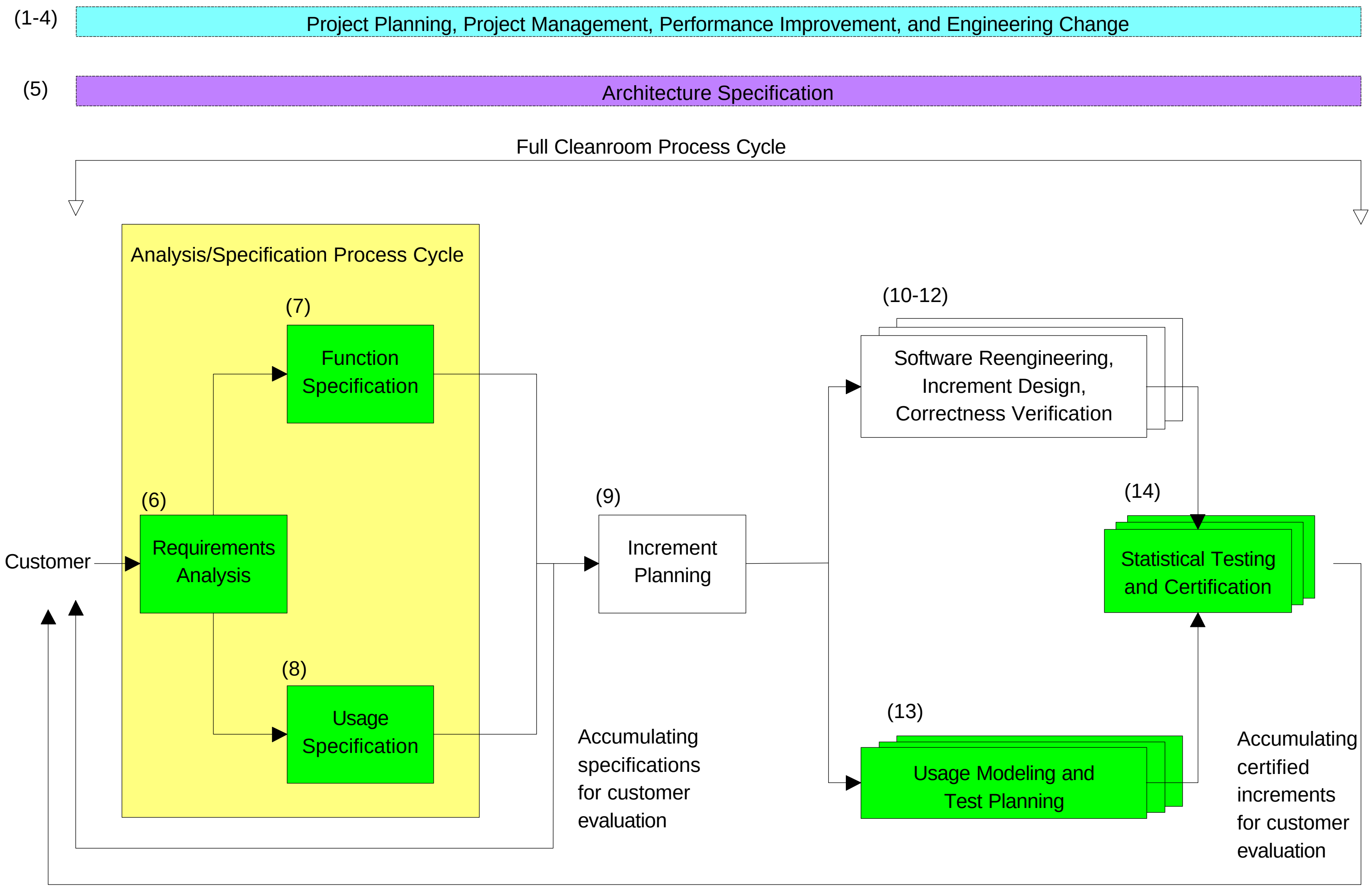
Software Development Processes

- Linear sequential
- Evolutionary
- RAD
- Prototyping
- **Formal - Cleanroom SE**
- 4GL

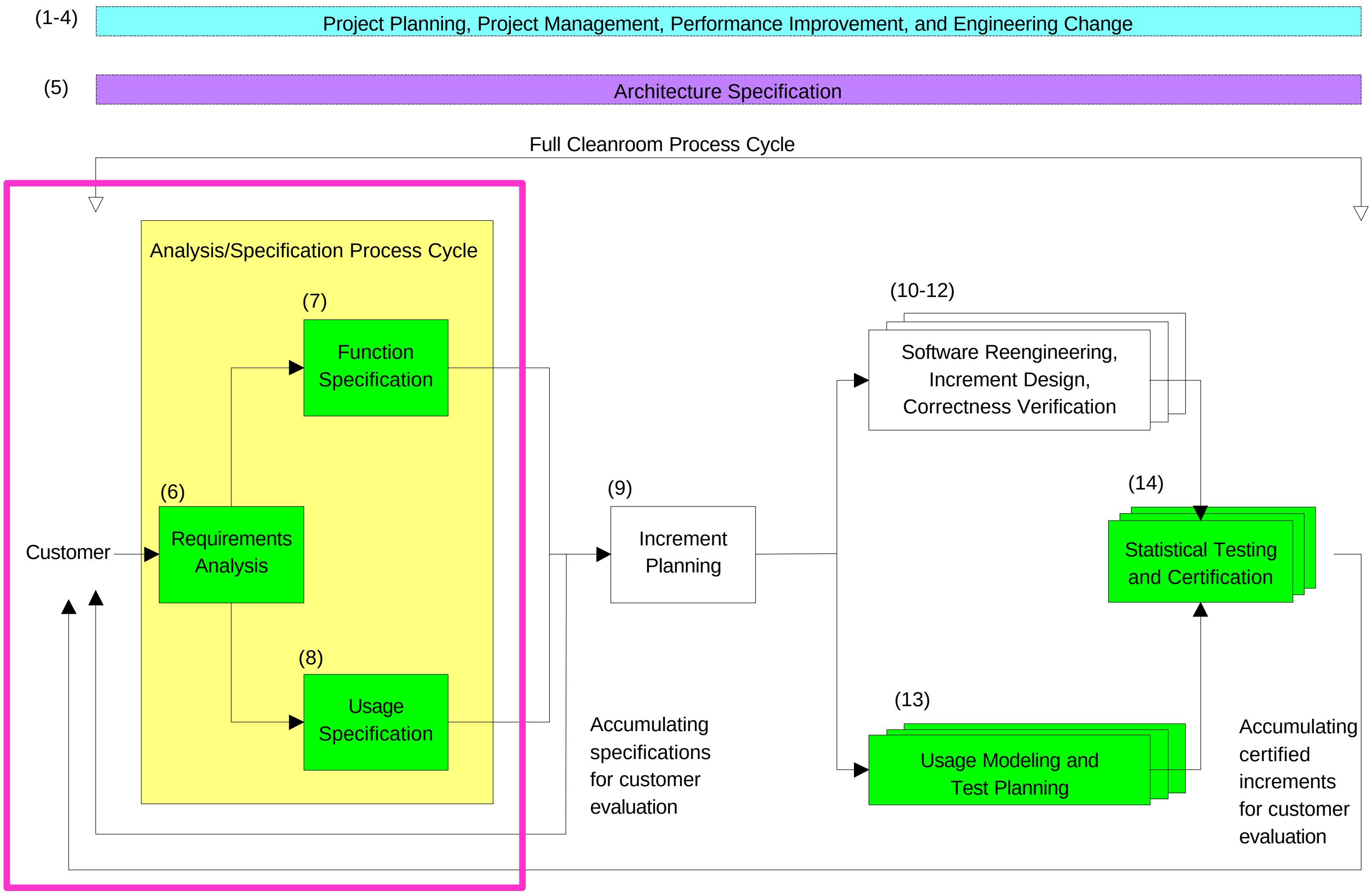
Cleanroom Software Engineering Process

- Theory based
- Team oriented

Cleanroom SE Reference Model



Cleanroom SE Reference Model



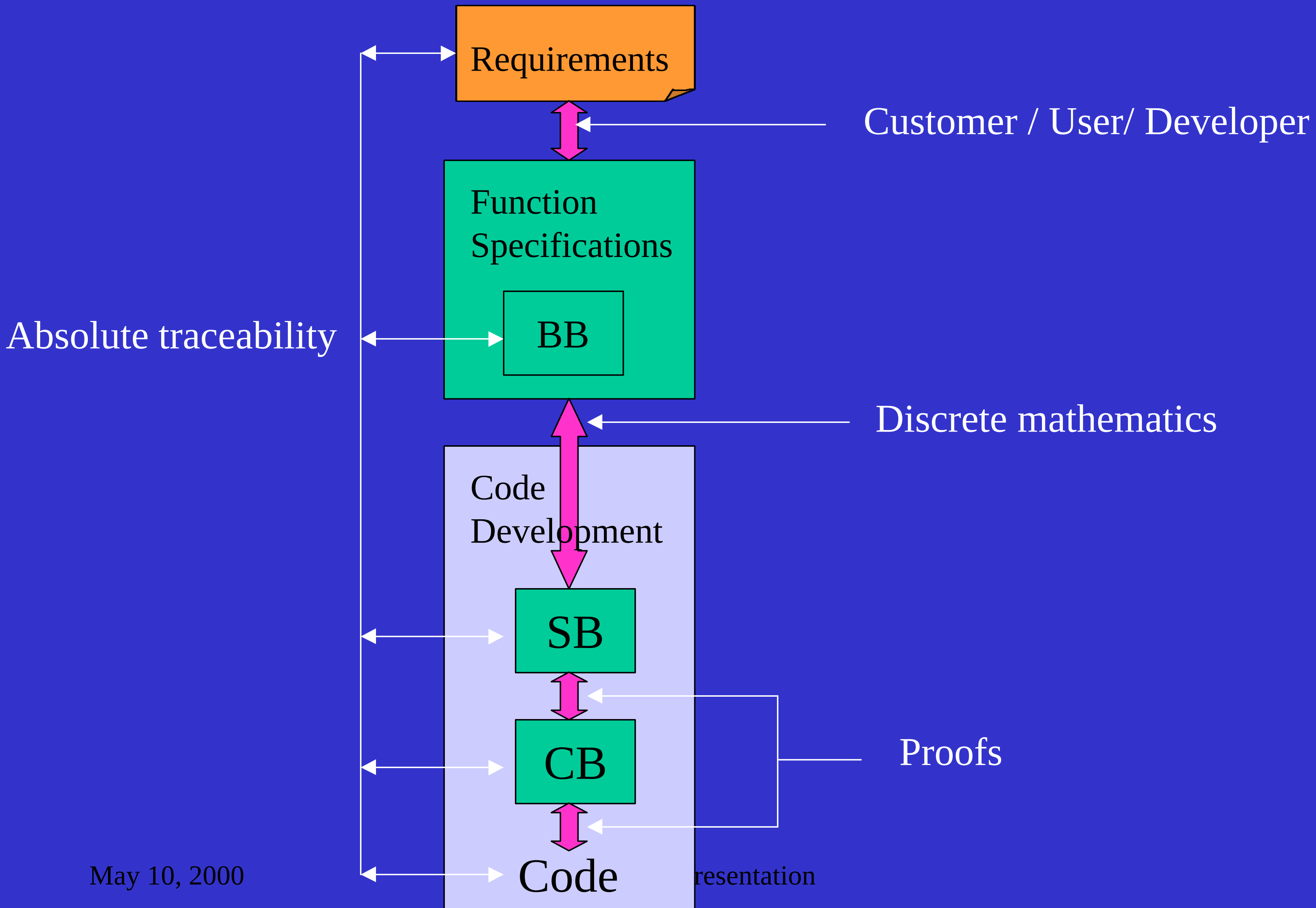
Specification Team Tasks

- Identify and record requirements
- Refine **requirements** and **risk assessment** to establish the **functional specification** thru enumeration
- Verify functional specification with "users"
- Define system architecture
- Develop **usage specification**

Code Development Team

- Begin with the functional (Black Box) specification and produce verified code in the language of choice
- Box structure design
  - Black Box → State Box
  - State Box → Clear Box
  - Clear Box → Code
- Verify steps in frequent team reviews

Referential Integrity



Requirement Example

The <u>printer driver</u> shall transfer files to a printer using XON **/** XOFF flow control. The printer driver is to perform as follows.

The computer sends data, one byte at a time, to the printer. Any information received from the printer is to be displayed on the screen for the user to read. Information coming from the printer takes precedence over information to be sent to the printer, so all bytes sent from the printer must be read before the next byte of data can be sent to the printer. (Information from the printer typically consists of error messages such as “no paper” or “paper jam.”)

If the printer sends the special byte XOFF the computer must stop sending bytes, and wait for the printer to send a subsequent XON byte. While waiting for XON, the computer should continue displaying any messages from the printer. Once XON is received, the computer can return to sending bytes (after first checking for any messages from the printer).

Requirements tagged and reorganized

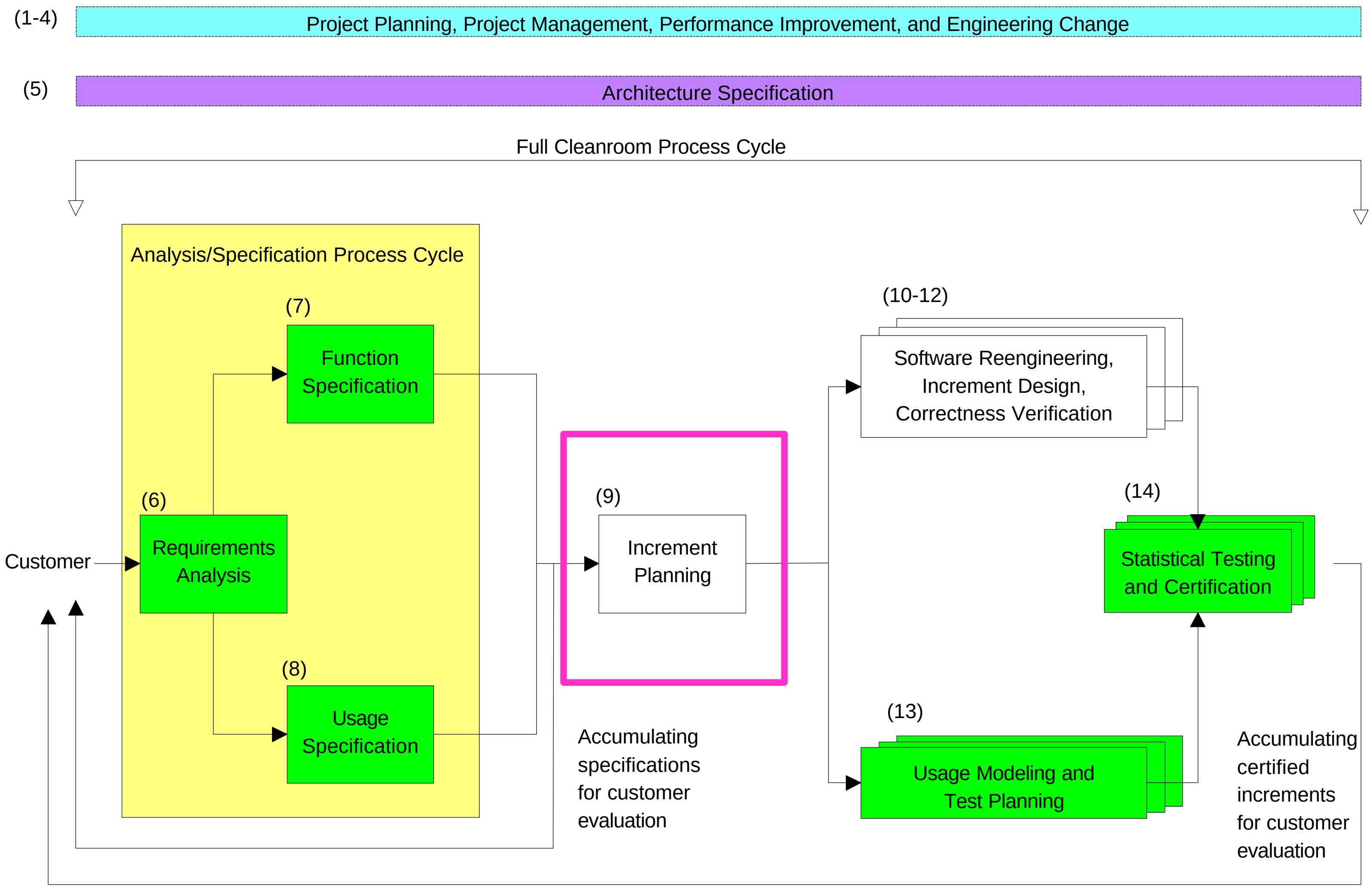
RqtID	Requirement	Trace
p1	**print file**	
p1.1	printer message (PMSG) takes precedence over sending data	
p1.2	printer does not send intervening XOFF's	
p1.3	when printer sends XOFF	
p1.3.1	computer must stop sending data	
p1.3.2	computer must wait for XON	
p1.3.3	computer must continue displaying any PMSG	
p1.4	when printer sends XON	
p1.4.1 **p**1.4.1.1 **p**1.4.1.2	computer checks for PMSG If PMSG then display PMSG on screen else remove <u>old</u> PMSG from screen Endif	q5
p1.4.2	send data from computer to printer one byte at a time	
p1.4.3	computer continues sending bytes until printer sends XOFF	

p1.3.3	computer must continue displaying any PMSG	
p1.4	when printer sends XON	
p1.4.1 **p**1.4.1.1 **p**1.4.1.2	computer checks for PMSG If PMSG then display PMSG on screen else remove old PMSG from screen Endif	q5
p1.4.2	send data from computer to printer one byte at a time	
p1.4.3	computer continues sending bytes until printer sends XOFF	
p1.4.4	computer continues sending bytes until EOF encountered	q3
p1.5	printer does not send intervening XON's	
p1.6	computer can't process EOF until printer sends XON	
p1.7	when computer receives an EOF no more XON, XOFF, or PMSG's are accepted from the printer	
p1.8	once the computer receives an EOF, subsequent EOF's are illegal until a file is again opened.	
p1.9 **p**1.9.1 **p**1.9.2 **p**1.9.3 **p**1.9.4	when computer receives an EOF print driver stops sending data print driver performs some "house cleaning" (close file) print driver signals printer that file transfer is complete print driver goes to "sleep"	q3
p1.10	printer goes to "sleep" when not servicing the printer	q4
p1.11 **p**1.11.1 **p**1.11.2 **p**1.11.3	when the print driver is "awakened" print driver checks on printer status print driver gets file to print print driver requests printer to begin printing	q6
pmsg1	PMSG1 = no paper	
pmsg2	PMSG2 = paper jam	

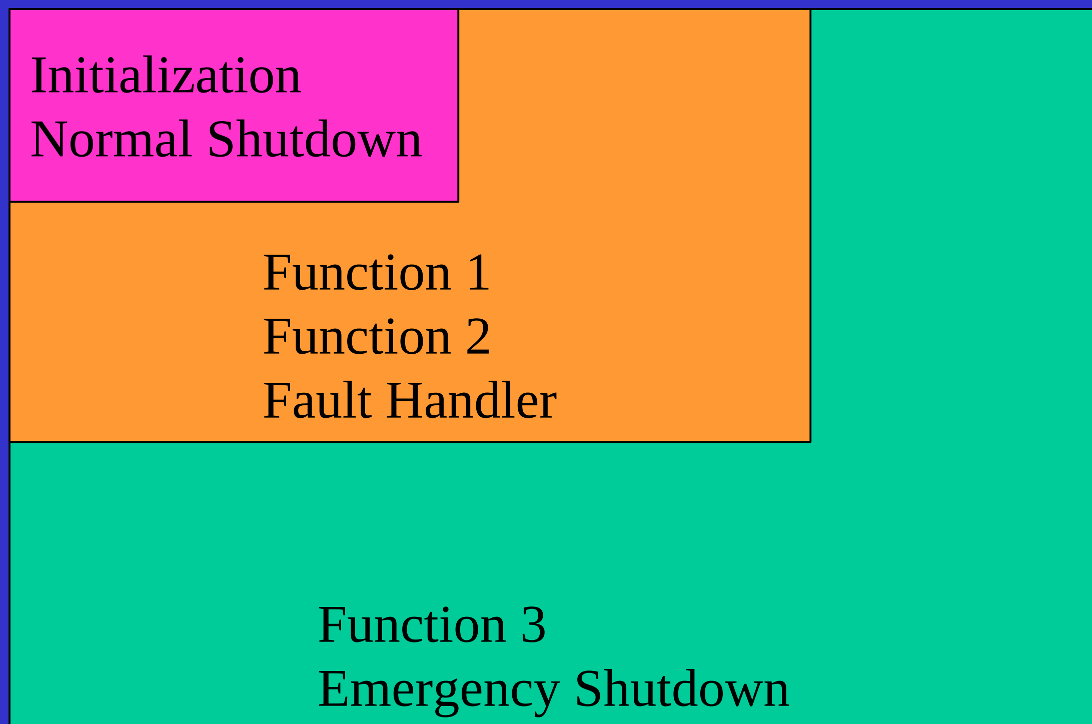
Questions & Derived Requirements

q1	what does the printer do when it runs out of paper?
q2	what does the printer do when the paper jams?
q3	what does the printer driver do when it receives an EOF? print driver stops sending data print driver performs some “house cleaning” (close file) print driver signals printer that file transfer is complete print driver goes to “sleep”
q4	what does printer driver do when not servicing the printer? print driver goes to “sleep” >> this is λ <<
q5	what does the print driver do after checking for a PMSG? If PMSG then display PMSG on screen else remove old PMSG from screen Endif
q6	what does the print driver do when it is awakened from its “sleep”? >> this is I << print driver checks on printer status print driver gets file to print print driver requests printer to begin printing

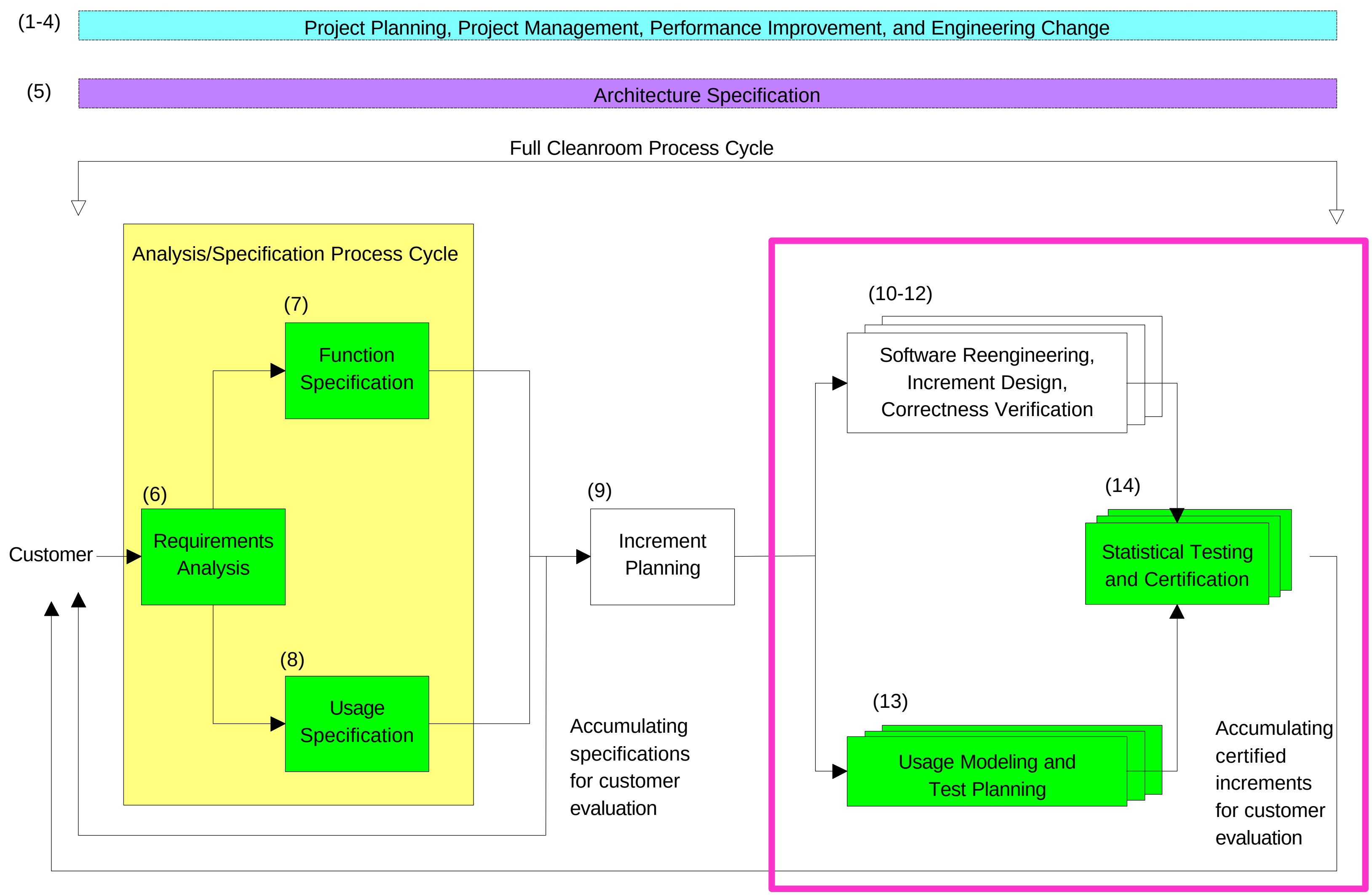
Cleanroom SE Reference Model



Increment Planning



Cleanroom SE Reference Model



Certification Team

- Compile code from developers
- Develop usage models from usage specification
- Verify the model with users
- Define test plan and validate with management
- Produce test scripts
- Run tests and compute results

Usage Modeling

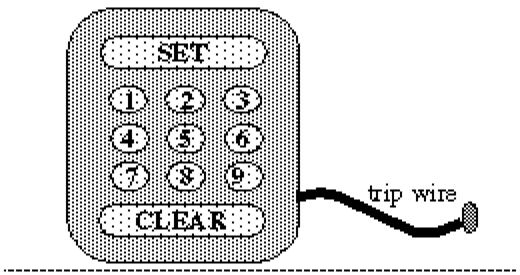
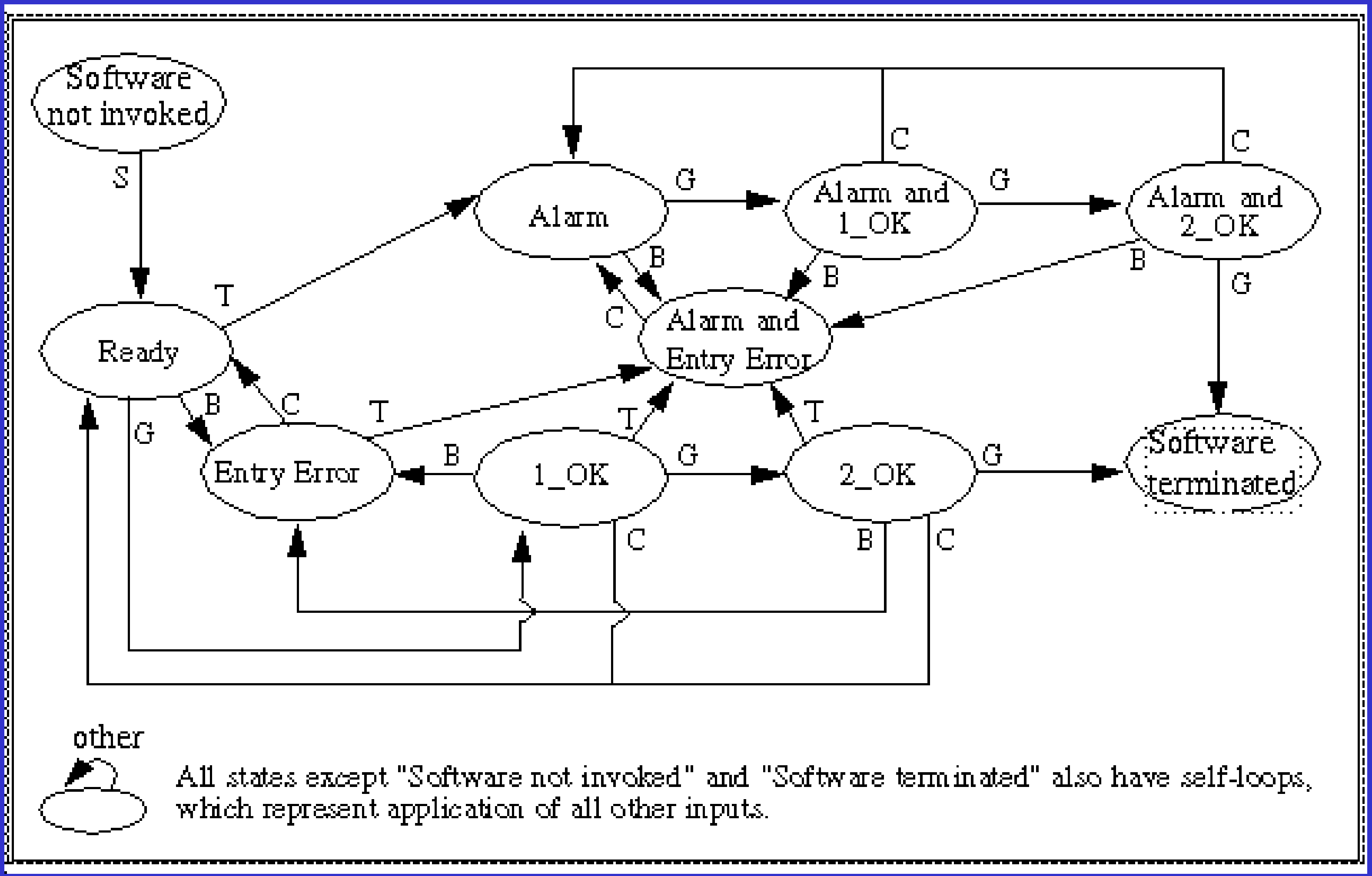


Table 2. Inputs for the X-Ray Cabinet Door Alarm

Source of Input	**Input**	**Description**
Detector	Trip (T)	Signal from detector
Human user	Set (S)	Device activator
	BadDigit (B)	Incorrect entry of a digit in the code
	Clear (C)	Clear entry
	GoodDigit (G)	A digit that is part of the three-digit deactivation code

Usage Modeling



Statistical Testing & Certification

Table 5. Measures of Test Sufficiency for the X-Ray Cabinet Door Alarm - no failures				
Script #	**Result**	**D(U,T)**	**% States Visited**	**% Arcs Traversed**
1	Pass	-	60.000	22.581
2	Pass	-	100.000	58.065
3	Pass	-	100.000	67.742
4	Pass	-	100.000	67.743

Statistical Testing & Certification

Table 8. Measures of product quality for the X-Ray Cabinet Door Alarm - one failure					
Script #	**Result**	**MTTF**	**Reliability**	**C=95%**	**C=99%**
1	Pass	---	1.000	0.000	0.000
2	Pass	---	1.000	0.000	0.000
3	Fail	3.000	0.667	0.252	0.331
4	Pass	4.000	0.750	0.198	0.261
5	Pass	5.000	0.800	0.163	0.214
6	Pass	6.000	0.833	0.137	0.180
7	Fail	3.500	0.714	0.140	0.184
8	Pass	4.000	0.750	0.124	0.163
9	Pass	4.500	0.778	0.112	0.147

Evidence for Regulators / Auditors

- Requirements are complete, current, and detailed
- Functional specification describes every possible response of the software and is readable
- All requirements are traceable to code

Evidence for Regulators / Auditors

- All of these data are available to evaluate the quality and completeness of tests
  - Test Plan
  - Usage Model
  - Coverage Test Results
  - Statistical test evidence (reliability data)
  - Critical use test results

Success Stories

- IBM mass storage control unit adapters
- SEL at NASA Goddard Space Flight Center
- U.S. Army Picatinny Arsenal

Overall Cleanroom Results

• Productivity Improvement	200-400%
• Quality Improvement	10-100:1
• Code size reduction	5:1
• Return on Investment	20:1

Adaptability

- Portions of a project
- Legacy system certification
- COTS certification
- Reverse Engineering
- Hardware test platform unavailable

OST Projects

- WRAIR infusion pump
- RTP safety model
- CBER regulatory policy model

Summary

- Formal Methods
  - Produce much better code
  - Produce much better documentation for review
  - Cost manufacturers less to build
  - Costs much less to review and approve
- Usage Model technology
  - Applies to any software
  - Can simplify the regulation of software