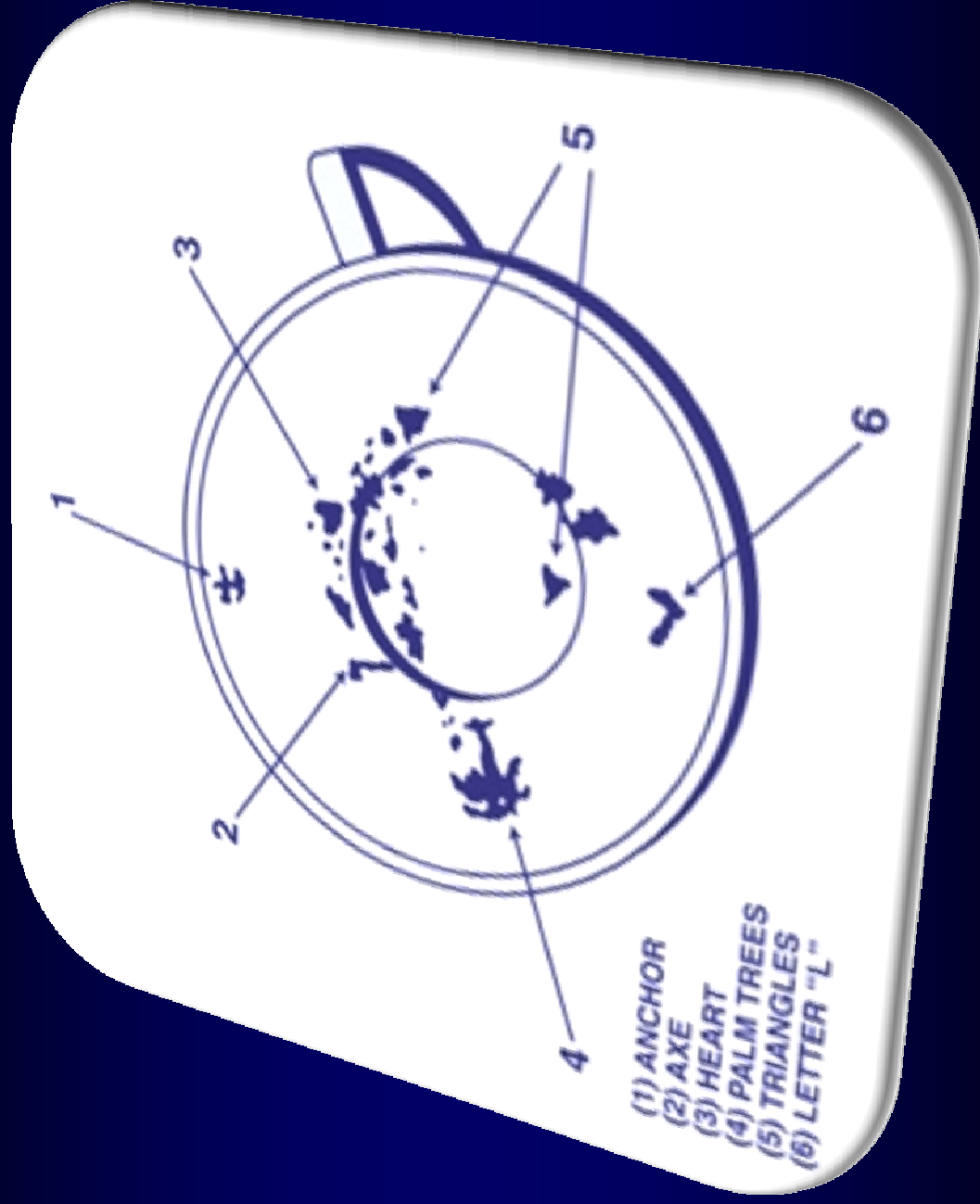
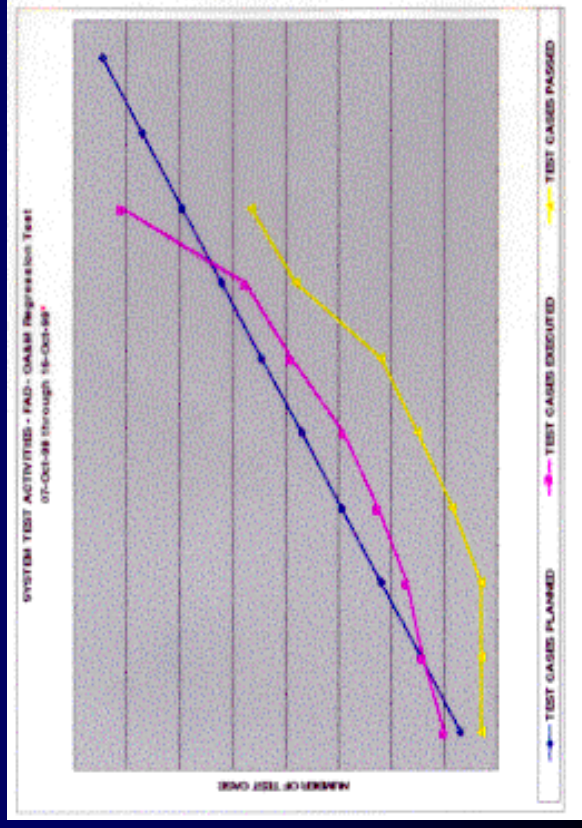
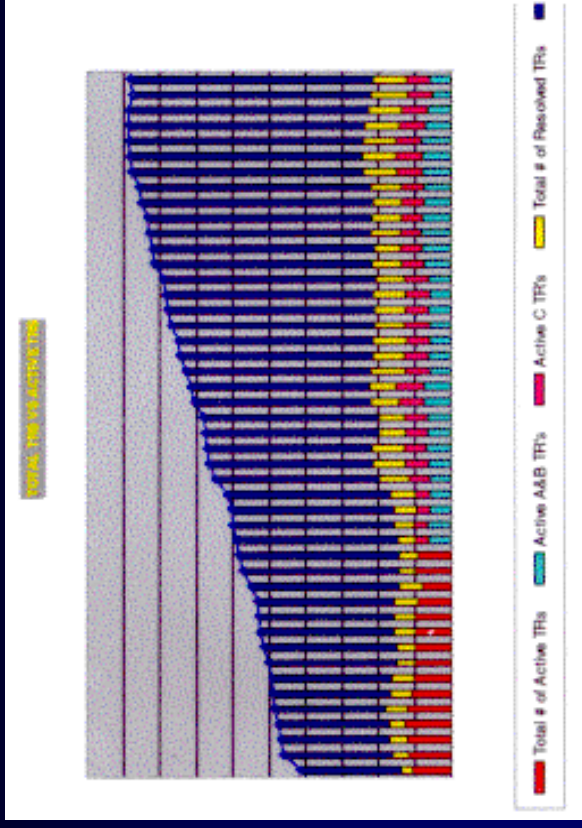


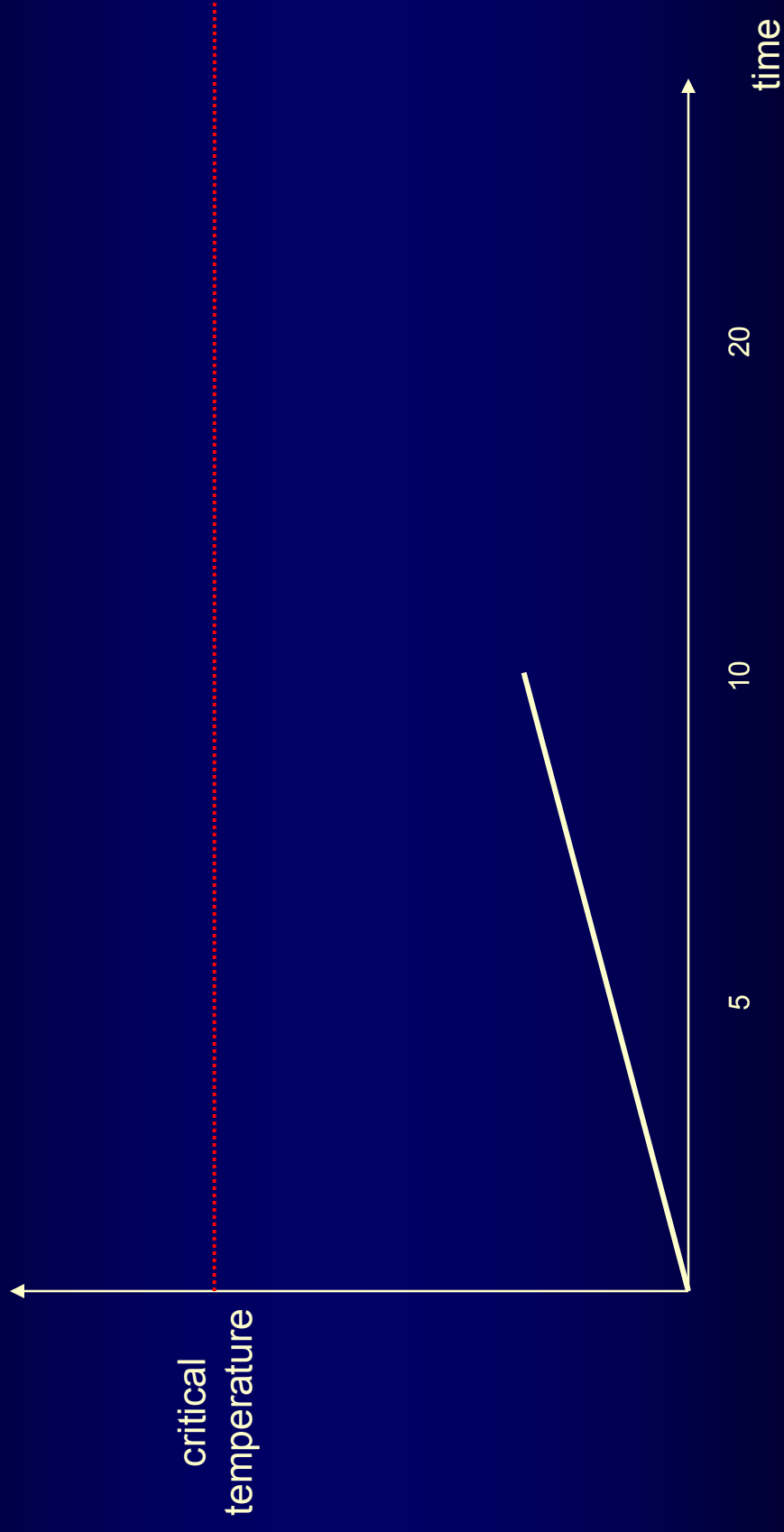


Key project manager questions

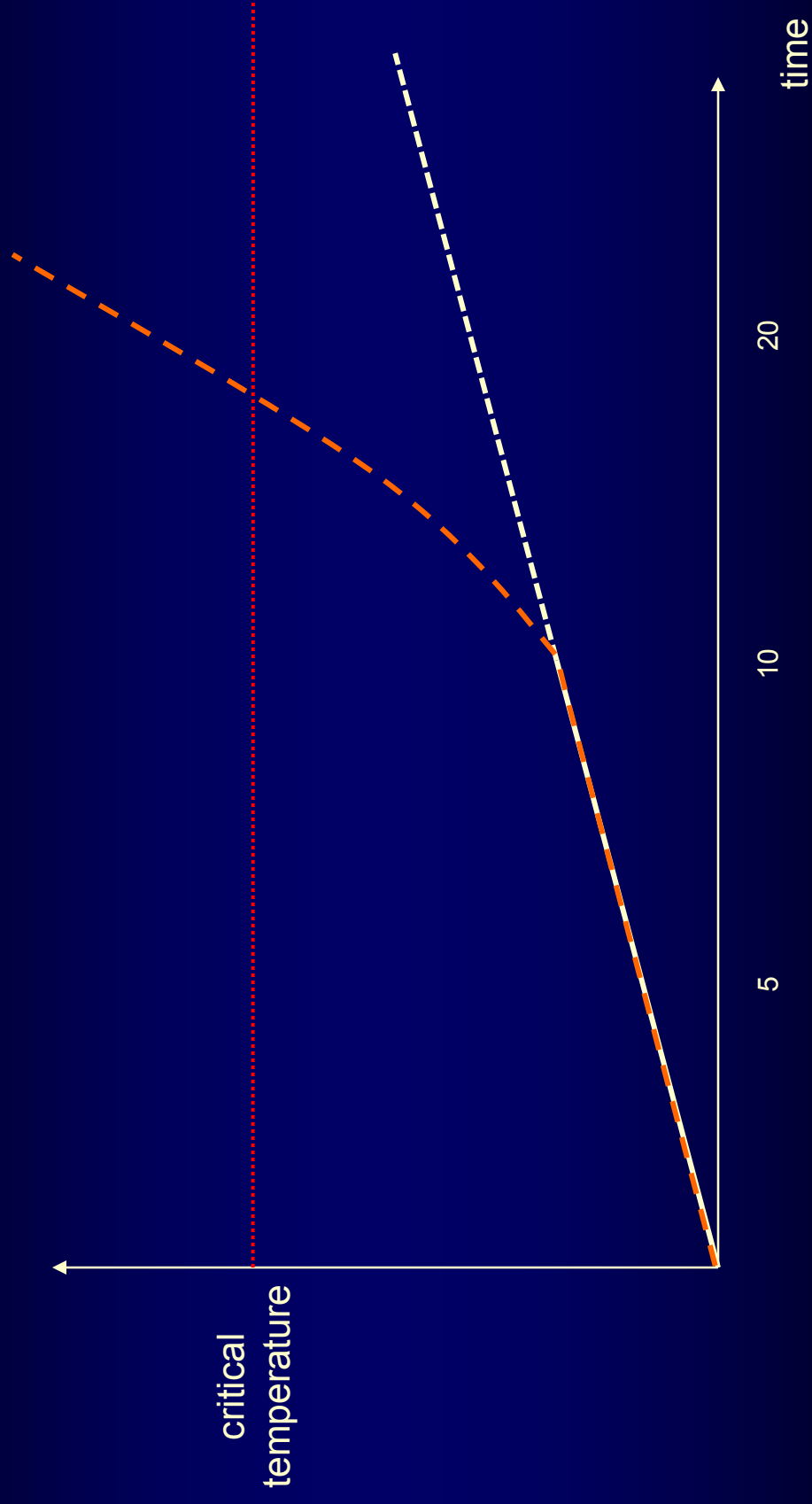


- How long more is going to take?
- How do I know the work is completed?
- Are we on track?
- When will it be ready for shipment?
- If we release it now, what should we expect?

What would you do?



What would you do?



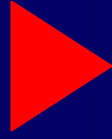
Understanding failure discovery patterns

“ ... There is a rhythm and a pattern between the phenomena of nature which is not apparent to the eye, but only to the eye of analysis; and it is these rhythms and patterns which we call Physical Laws... ” ...look at the thing,...record the details, and...hope that in the information thus obtained might lie a clue to one or another theoretical interpretation.” ...Physics is not mathematics, and mathematics is not physics. One helps the other. But in physics you have an understanding of the connection of words with the real world.”

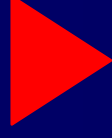
R. Feynman, Physics Nobel Prize 1967

Single release patterns

Software
is
released



Testing
ends

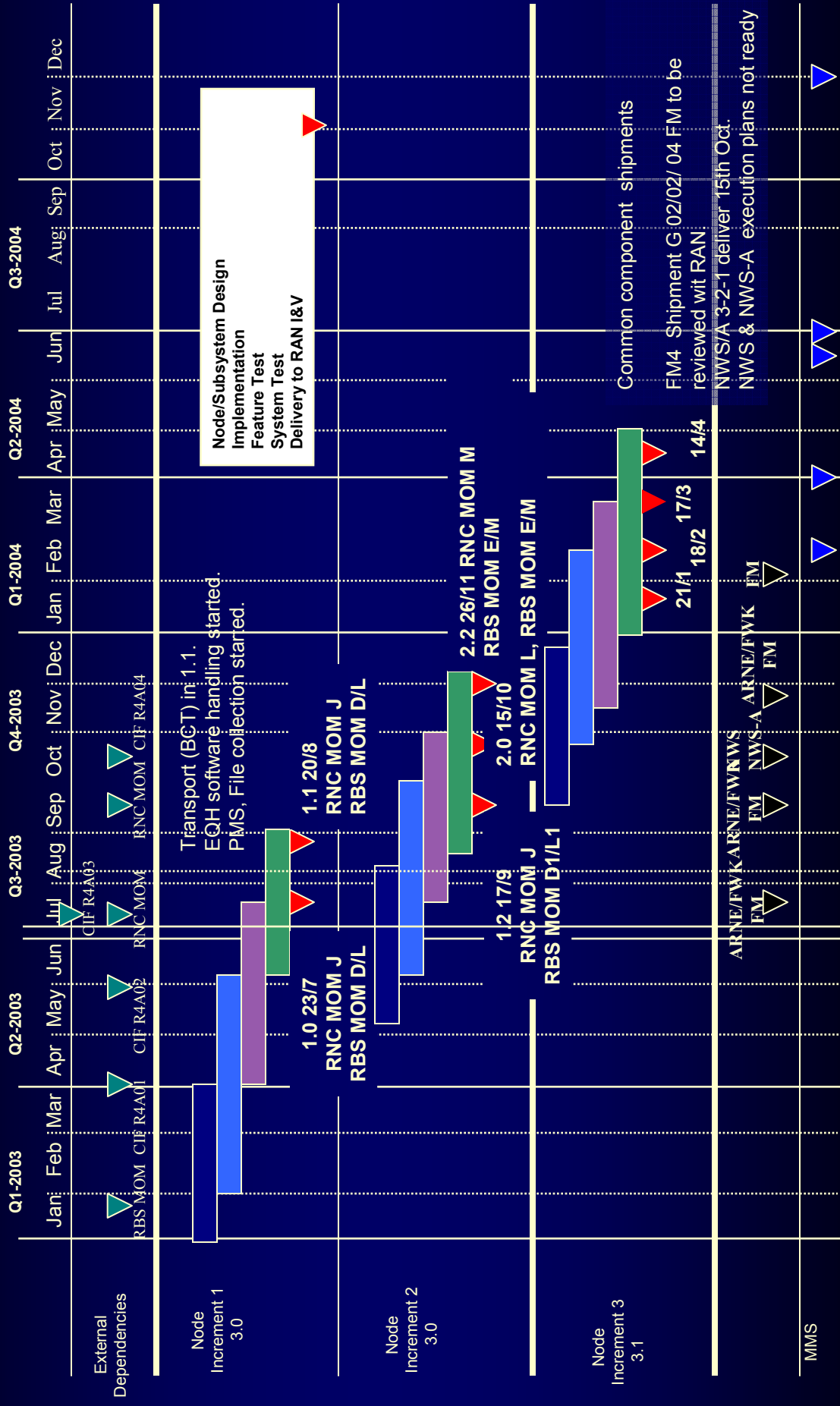


Testing

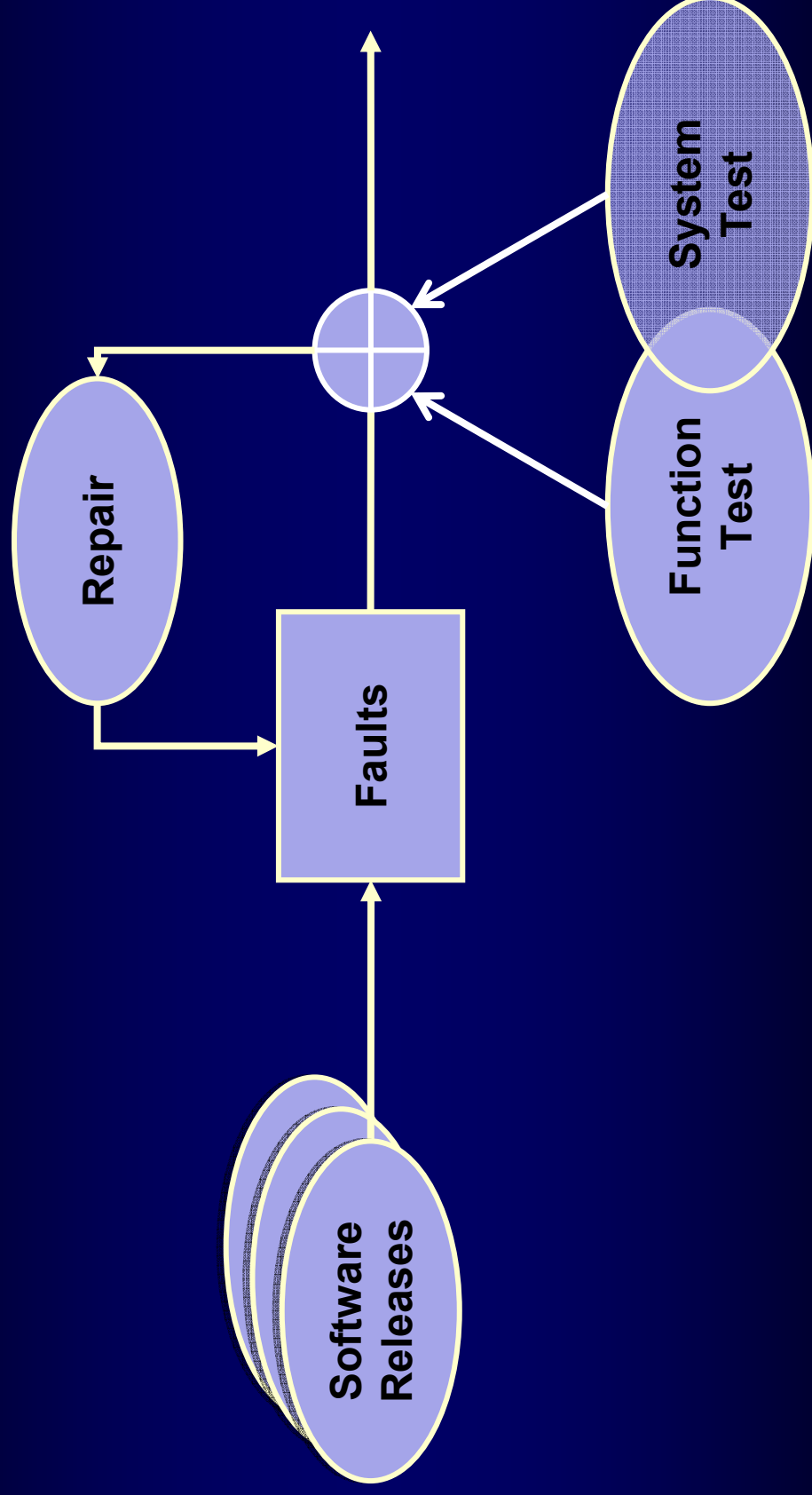
Things to be aware

Introductions	● Testing is a random process, so the models will work most of the time, but not all the time
Motivation for the course	● The application of any model is not a mechanical process. It requires knowledge of the underlying process
Focus of the course	● Models do not work well with small numbers.
Selecting the tools	● Software with few faults or sparse testing tend to fail to exhibit any failure discovery pattern. Interval predictions tend to be less accurate towards the right side of the distribution
Cost of quality	● The models only tell you what is the trend the data conforms to. Whether this is a problem or not, is for you to decide. If there is a problem, the models do not provide information about the causes
Life cycles	
Long term consequences	
Good work	
Administration	
Questions	

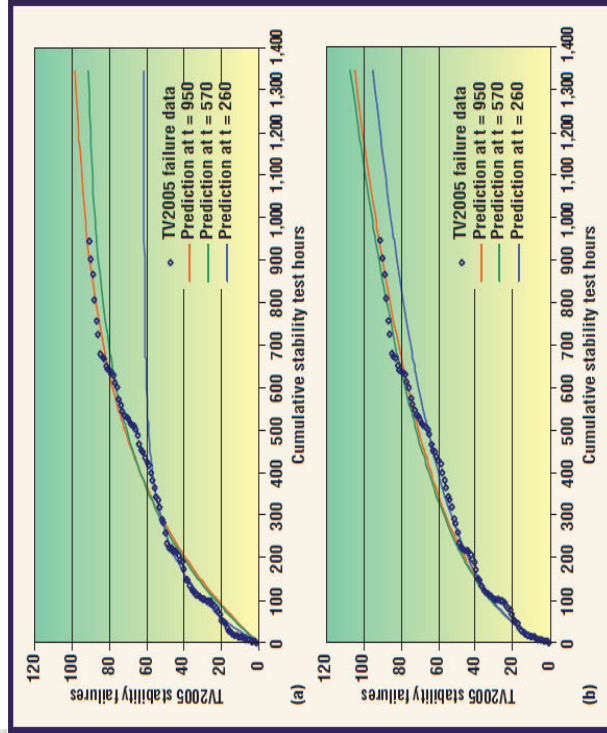
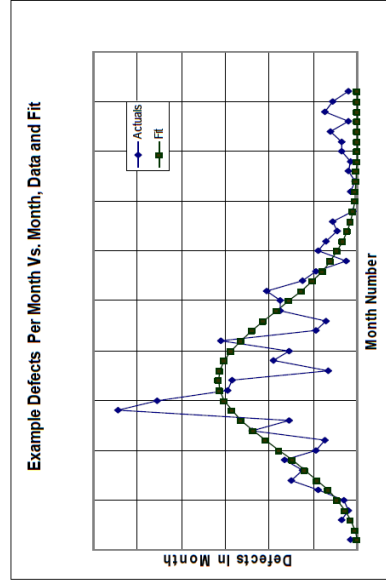
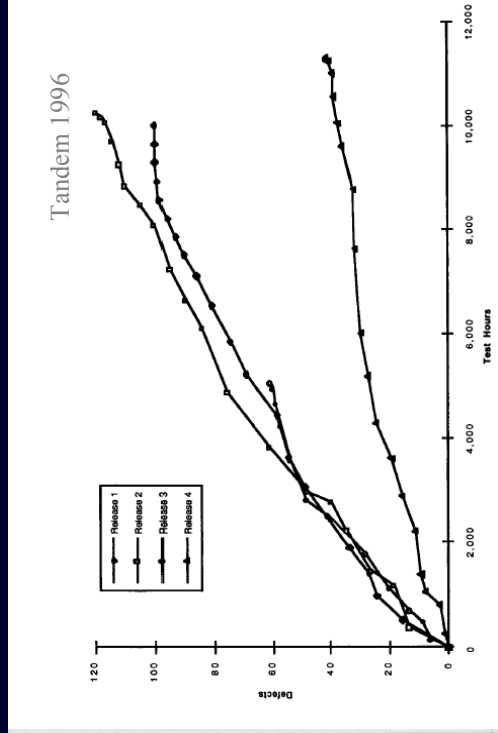
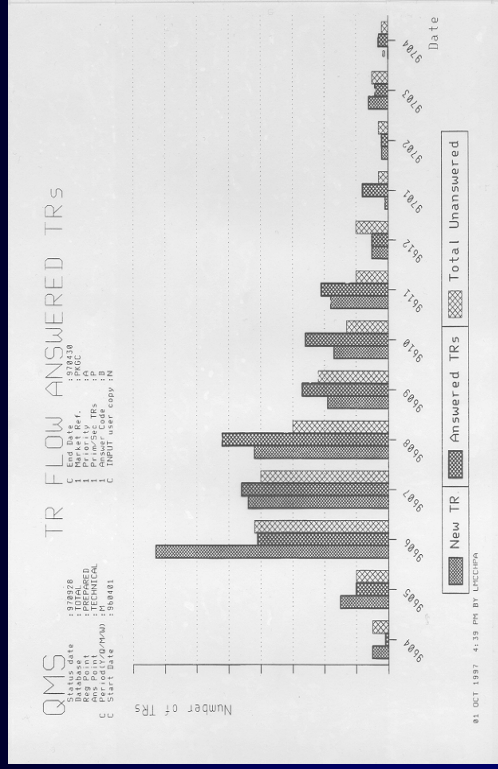
What happens when the process gets a little more complicated?



A Testing Process Model



Very nice, but does it work in practice?



1. Ericsson 1998
2. Lockheed Martin 2003
3. Tandem 1996
4. NPX 2008

How do we use these patterns to answer the key management questions?

- How long more is going to take?
- How do I know the work is completed?
- Are we on track?
- When will it be ready for shipment?
- If we release it now, what should we expect?

Data used as input to the EPM Tool

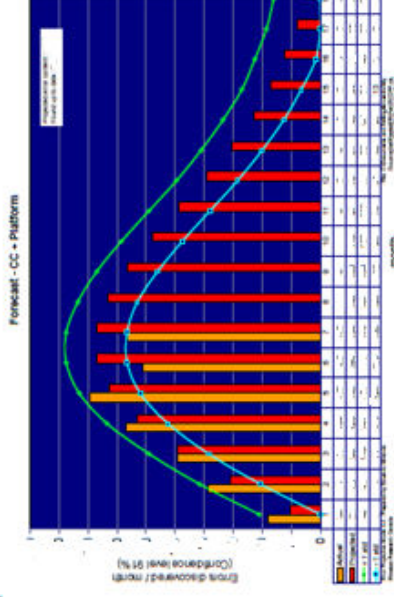
Month	Platt. + CC	WCDMA	GSM	CORE
Jul-03	2.0	2.0		
Aug-03	2.0	2.0		
Sep-03	2.0	2.0		
Oct-03	2.0	2.0	2.0	
Nov-03	2.0	2.0	2.0	2.0
Dec-03	2.0	2.0	2.0	2.0
Jan-04				

* These values are the sum of all faults reported previously to the assumed start date for the beginning of the testing effort.

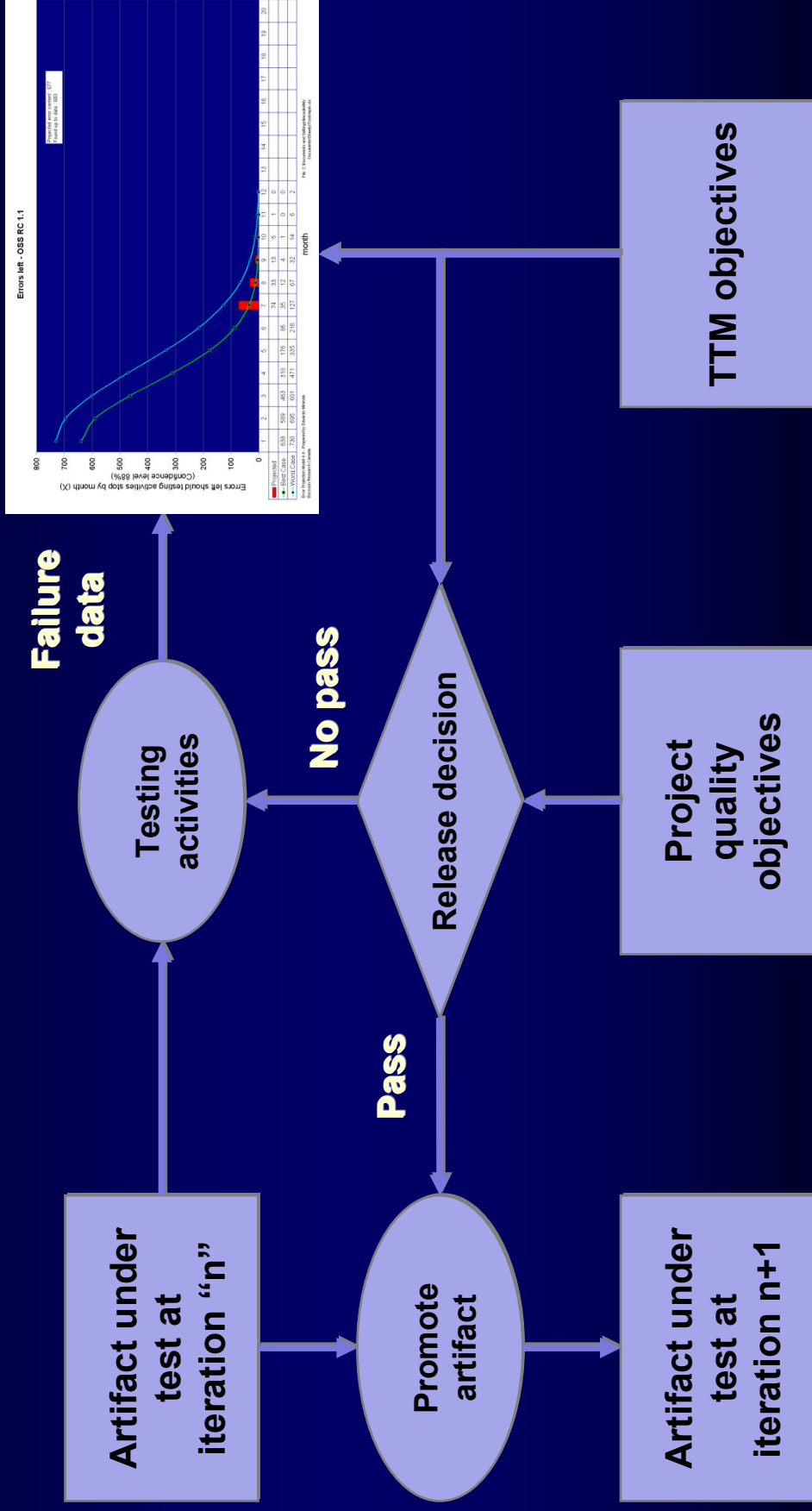
Common Components + Platform

Based on the amount of testing effort and the delivery of functionality Project Management for the R2 project puts the peak of the fault discovery curve at December 2003. This seems to be consistent with the data that shows a decline on the influx of registered faults starting in November 2003. See chart below.

The total fault content forecasted at this point is 100, of which 10 have already been found.



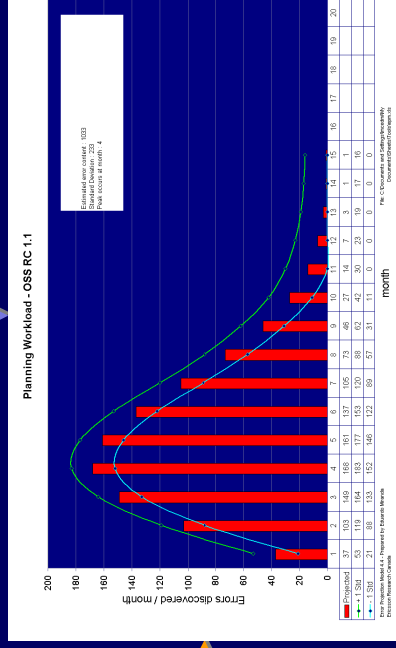
Preventing the premature promotion of software builds



Planning testing activities

Past Projects,
own
experience

A priori estimation of
error content &
quality objectives

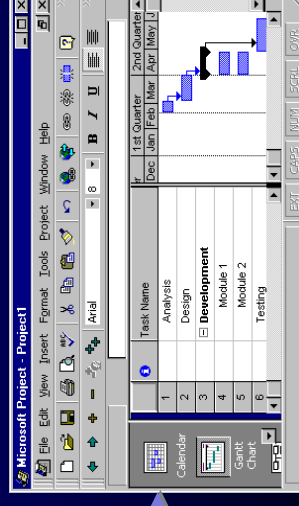


Testing duration

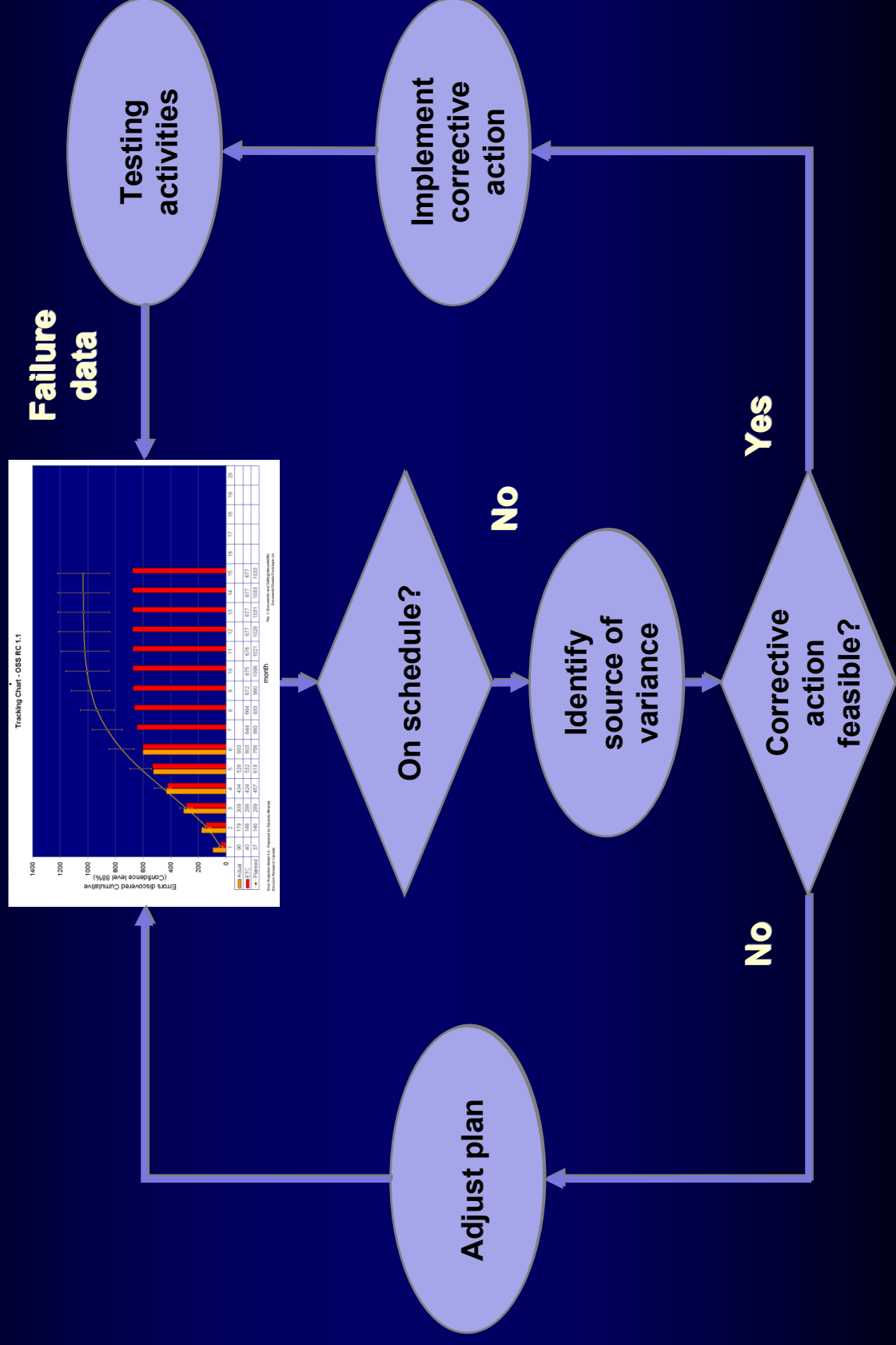
Yes

Can you handle it ?

No



Monitoring the progress of testing activities



Questions?