The Formula for Success in System Testing

Nathan Petschenik
STS Consulting, Inc.
732-257-9033 x12
nathan@stsv.com

WBE Certified   GSA Contract GS-35F-0108T

Biography

Nathan Petschenik is Chief Operating Officer of Software Testing Services, Inc., an IT consulting firm specializing in all aspects of software testing, quality assurance, and compliance. In recent years Nathan has been focusing on Independent Verification and Validation (IV&V) across multiple industries. This includes Computer System Validation for clients in FDA regulated environments. In addition to the Life Sciences, his background includes technical and management consulting to clients in industries such as telecommunications, financial services, and government. Nathan's career in System Testing leadership has also included significant experience as a software developer, software development manager, systems engineering director, product manager, and project manager. He is the author of many technical papers that have appeared in professional journals and at technical conferences. Nathan's paper "Practical Priorities in System Testing", which was first published in IEEE Software, is now considered a landmark paper in the transition of software testing from Art to Science. His book "System Testing with an Attitude" was published by Dorset House in 2005. Nathan holds a Master of Science from Stevens Institute of Technology and has been certified as a Function Point Specialist.

The Formula for Success in System Testing

System Testing Success =

(Technical Excellence)

+

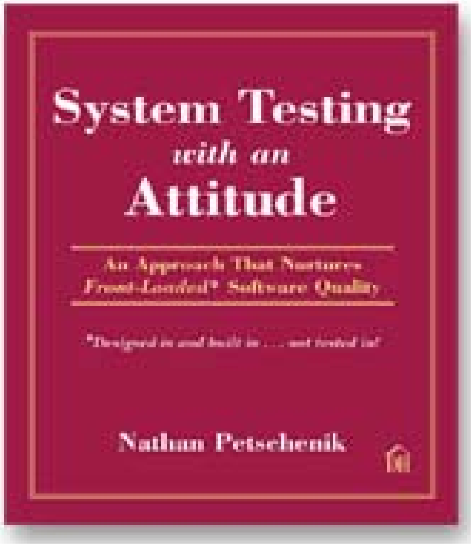
(Nurturing Front-Loaded Quality)

where:

(Technical Excellence) >0

(Nurturing Front-Loaded Quality) >0

Main Source


System Testing
with an
Attitude
An Approach That Nurtures
Front-Loaded® Software Quality
"Designed in and built in . . . not tested in"
Nathan Petschenik

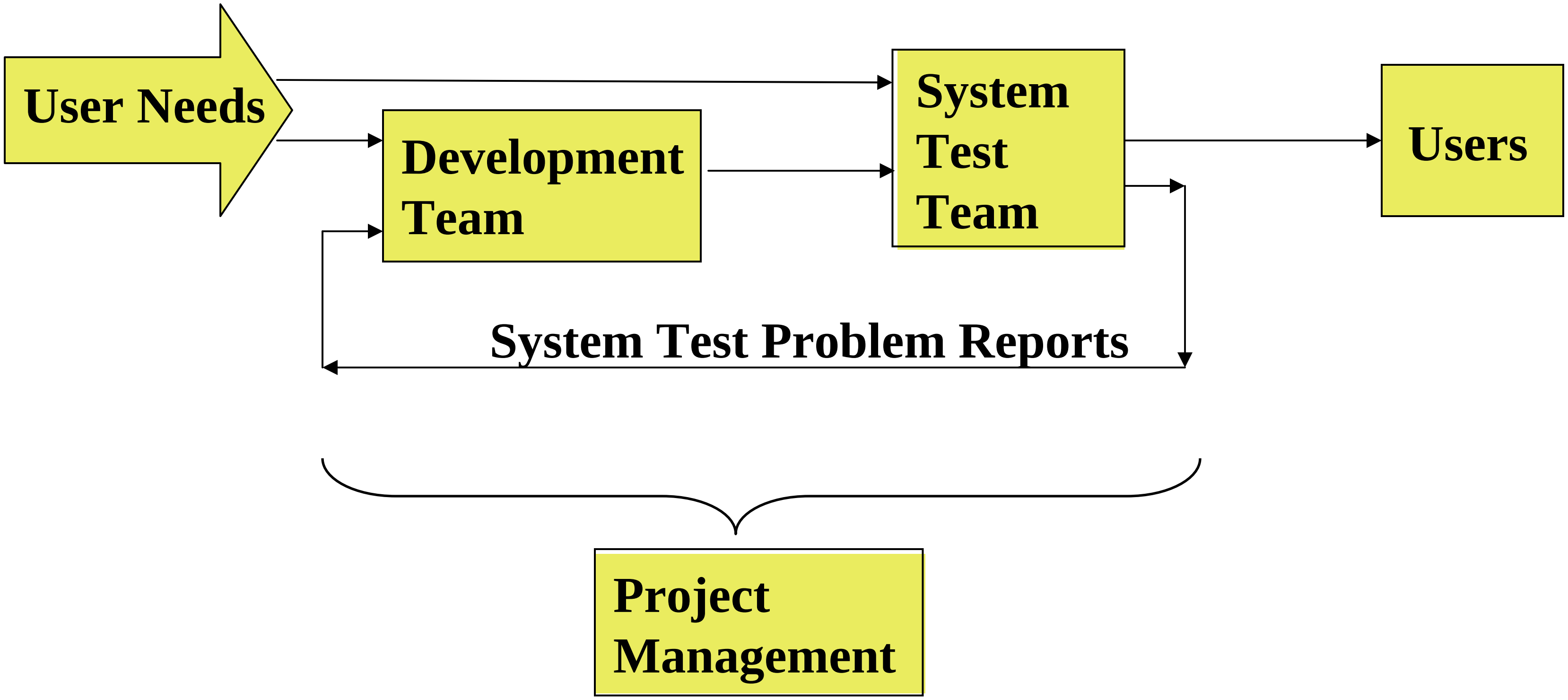
System Testing with an Attitude

An Approach That Nurtures
Front Loaded Software Quality

by Nathan Petschenik

ISBN: 0-932633-46-3
Dorset House, New York
©2005

Model of System Testing



User Needs

Development Team

System Test Team

Users

System Test Problem Reports

Project Management

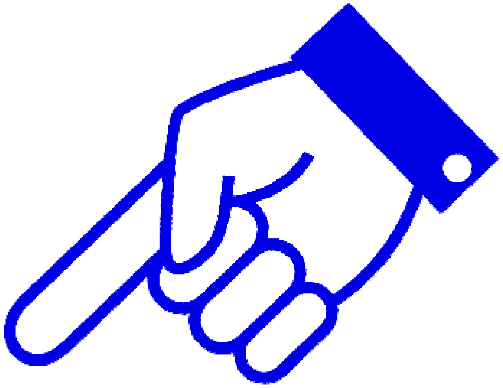
Misunderstanding Between Developers and Testers

I can’t wait to get feedback on whether my software works!

Developer

This software looks like it was never tested!

System Tester



How could you have missed that problem?

Role Awareness Seminars

Objectives:

- Build awareness of system testing issues
- Clarify expectations
- Change attitudes
- Influence behavior

Technique:

- Small groups of developers, system testers, and project managers
- 12 True/False Questions
- Open discussion of answers

Sample Questions from Role Awareness Seminar

The role of the System Test Team is:

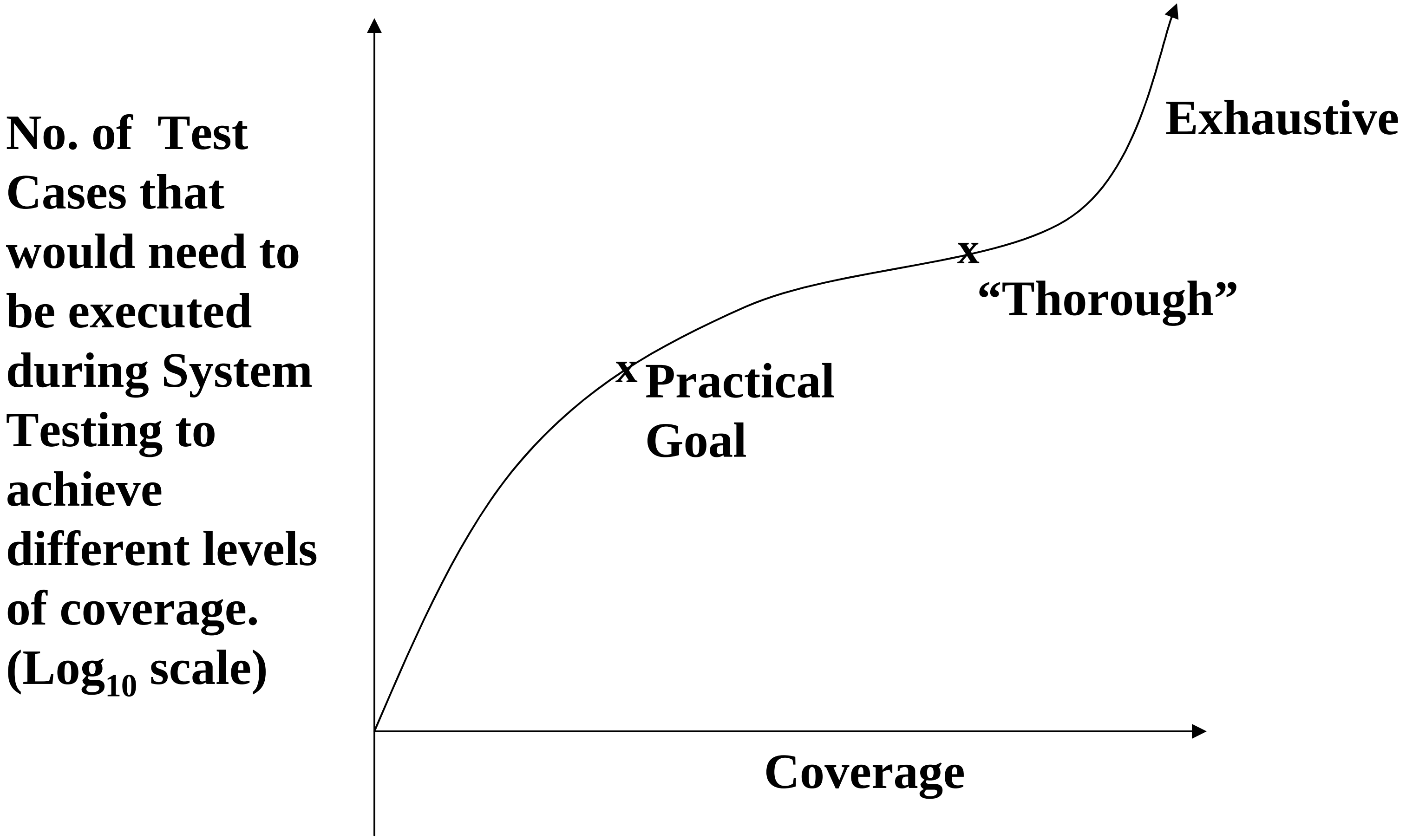
to supplement the testing done by the developers so as to achieve *exhaustive* coverage of the system.

T ( ) F( )

to test and retest the system *thoroughly* during the system test phase of the project.

T ( ) F( )

Levels of System Testing



Selecting Test Cases for System Testing

System testers select test cases using factors not considered in white box and black box coverage methodologies

- System Testers are looking for “important” problems that must be fixed before the system is provided to users

- System Testers are looking for the types of problems that may have been missed in the testing performed by developers prior to System Testing

Practical Priorities in System Testing

Rule A: Testing the system's capabilities is more important than testing its components.

Rule B: Testing old capabilities is more important than testing new capabilities.

Rule C: Testing typical situations is more important than testing unlikely situations.

Sample Question from Role Awareness Seminar

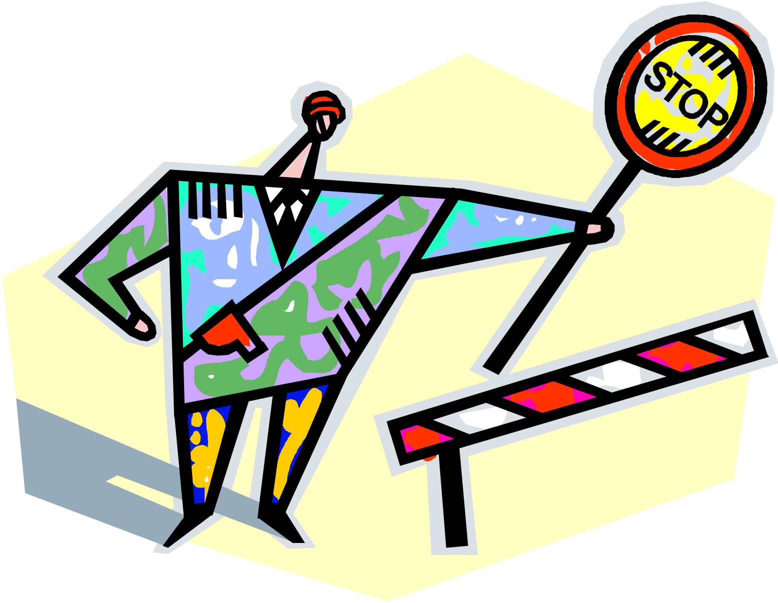
The role of the System Test Team is:

to help developers complete their testing.

T ( ) F( )

My Entrance Criterion to System Test

Developers must believe that the software is ready for users on the day that the software enters the system test phase



Obstacles to Meeting My Entrance Criterion

- Staff?
- Schedule?
- Lack of a stable integration testing environment?
- Lack of training in testing?
- Tools?
- Holes in Requirements?

Results of Breaking Down Real and Perceived Barriers to Quality

- Better developer testing
- Higher quality software to System Test
- Higher quality software to Users
- Productivity increases
- Non-tangible benefits

Ways for a System Test Leader to Nurture Front Loaded Quality

- Conduct Role Awareness Seminars (and stimulate action on follow-up items)
- Formalize Entrance Criteria to System Test
- Infiltrate the SDLC
- Champion the need for static testing techniques throughout the SDLC
- Be the collector and interpreter of measurements
- Become a role model for accountability

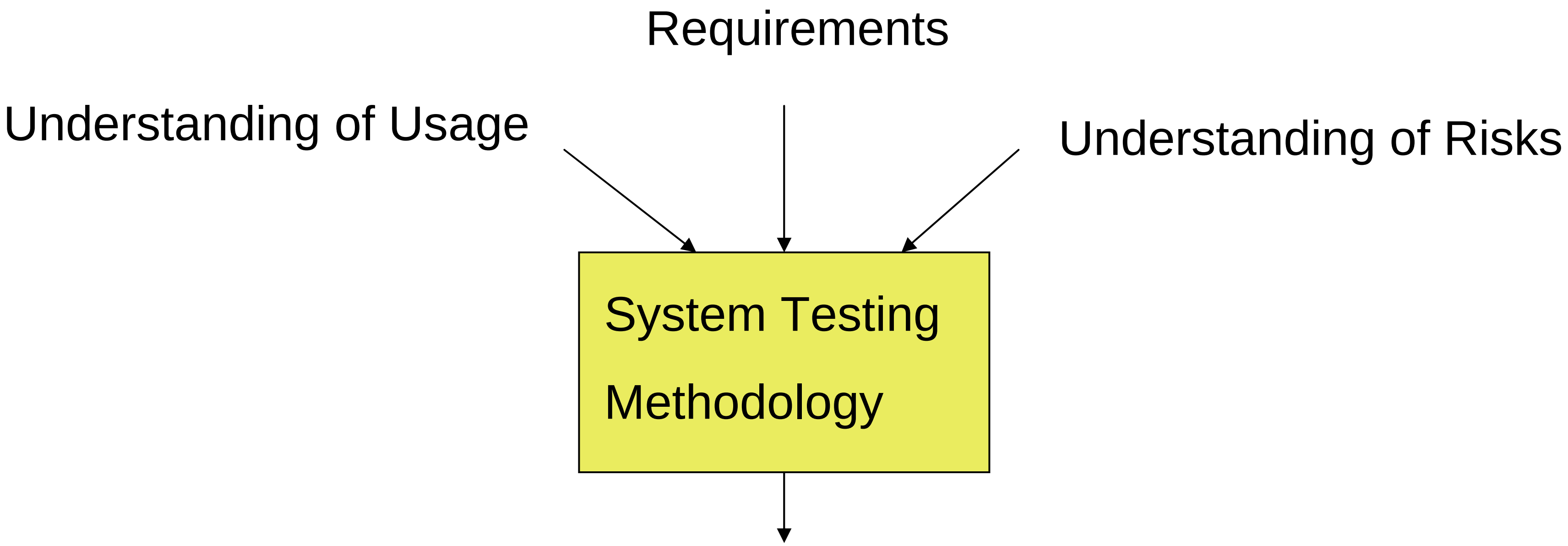
What does it mean for a System Test Team to Achieve Technical Excellence?

- Team consistently prevents important problems from reaching users
- Team members share a common approach, process, and/or methodology for addressing the technical challenges of system testing

Elements in the Science of System Testing

1. What system tests need to be developed?
2. How do you design system tests?
3. How do you make a test repeatable?
4. How do you solve cycle acceleration problems?
5. How do you test transactions-in-progress?
6. How do you leverage an existing test to cover additional data variations?
7. What should you measure?

1. What system tests need to be developed?



Requirements

Understanding of Usage

Understanding of Risks

System Testing

Methodology

- Architecture of System Tests
- Coverage responsibilities of each System Test in the Architecture (Test Objective)

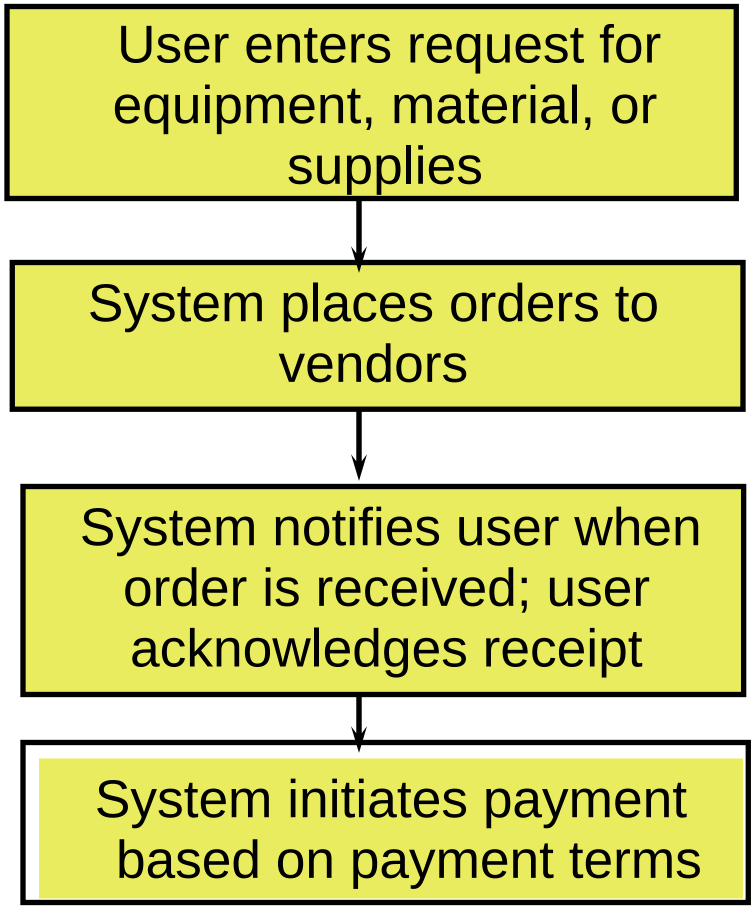
Understanding Usage: Typical Business Flow

Step 1 – Requisition

Step 2 - Order Placement

Step 3 - Receipt of Goods

Step 4 - Payment



User enters request for equipment, material, or supplies

System places orders to vendors

System notifies user when order is received; user acknowledges receipt

System initiates payment based on payment terms

Understanding Risks:
Risk Analysis

Function	Sub-Function	Risk Scenarios (Events)	Effect (s)	Risk Mitigation
Acquisition process	Entering Requisition	User requests non-approved supplier	Negative variations in quality could result in: • Safety hazard • Defective products • Fines • Human injury • Loss of life	Add management control features to enforce extra level of authorization on vendor deviation

Practical Priorities in System Testing

Rule A: Testing the system's capabilities is more important than testing its components.

Rule B: Testing old capabilities is more important than testing new capabilities.

Rule C: Testing typical situations is more important than testing unlikely situations.

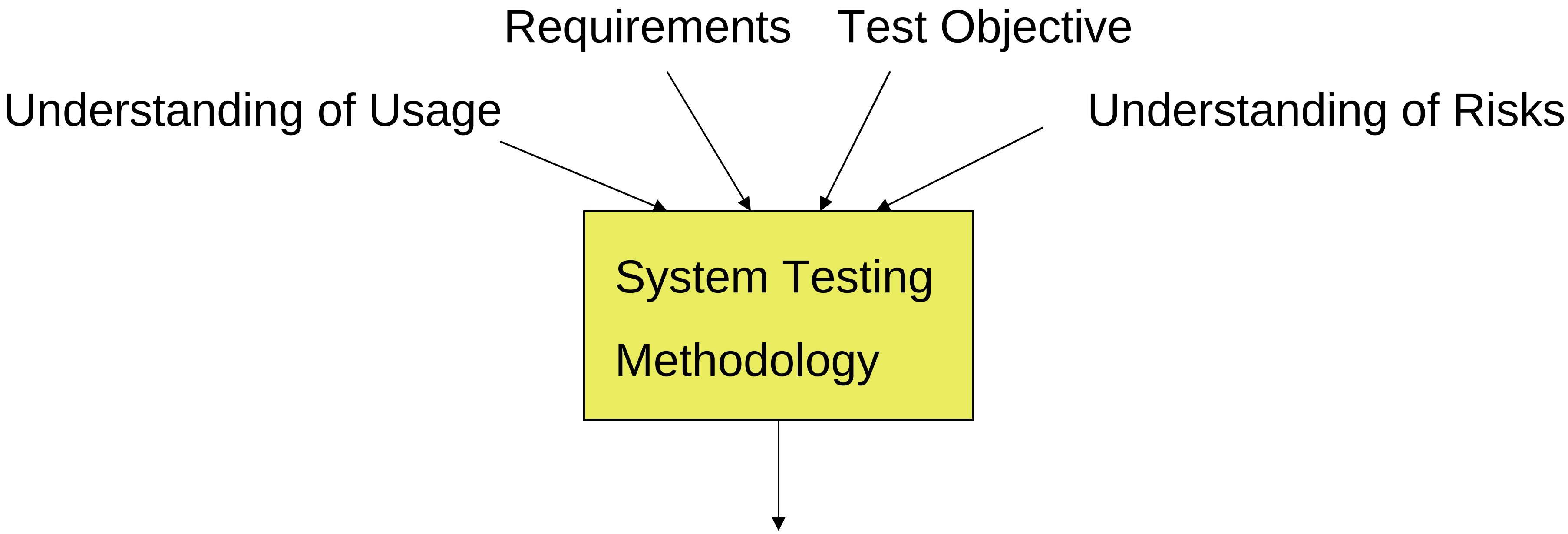
Importance of Risk Analysis in System Testing

Function	Sub-Function	Risk Scenarios (Events)	Effect (s)	Risk Mitigation
?	?	?		

Importance of Risk Analysis in System Testing (cont.)

Function	Sub-Function	Risk Scenarios (Events)	Effect (s)	Risk Mitigation
?	?	?		Fail safe controls

2. How do you design System Tests?



Requirements

Test Objective

Understanding of Usage

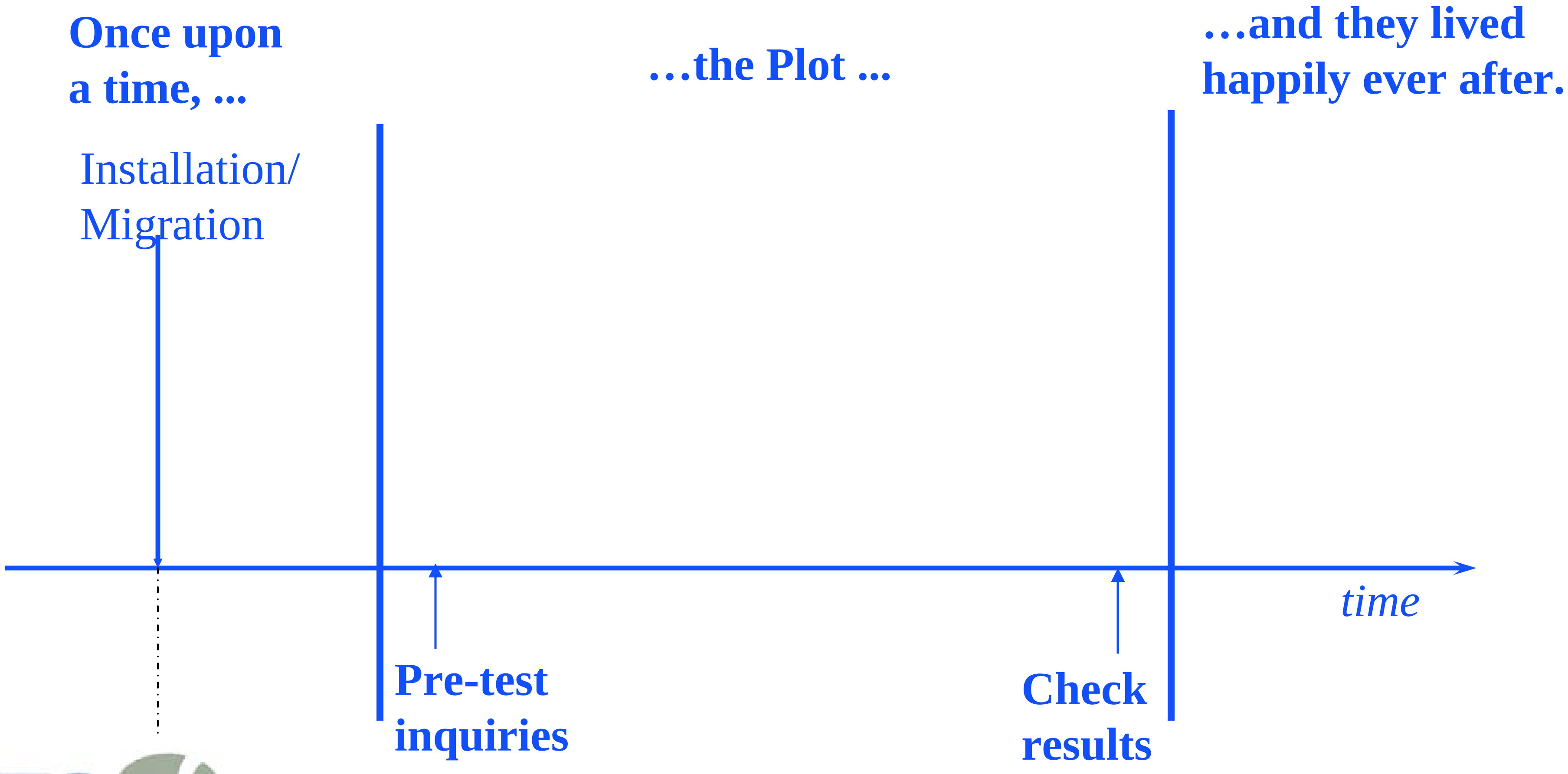
Understanding of Risks

System Testing Methodology

- Overall Design of System Test
- Test Design Documentation

2. How do you design System Tests? (continued)

Template for Designing the Story of the Test



Once upon a time, ...

…the Plot ...

…and they lived happily ever after.

Installation/ Migration

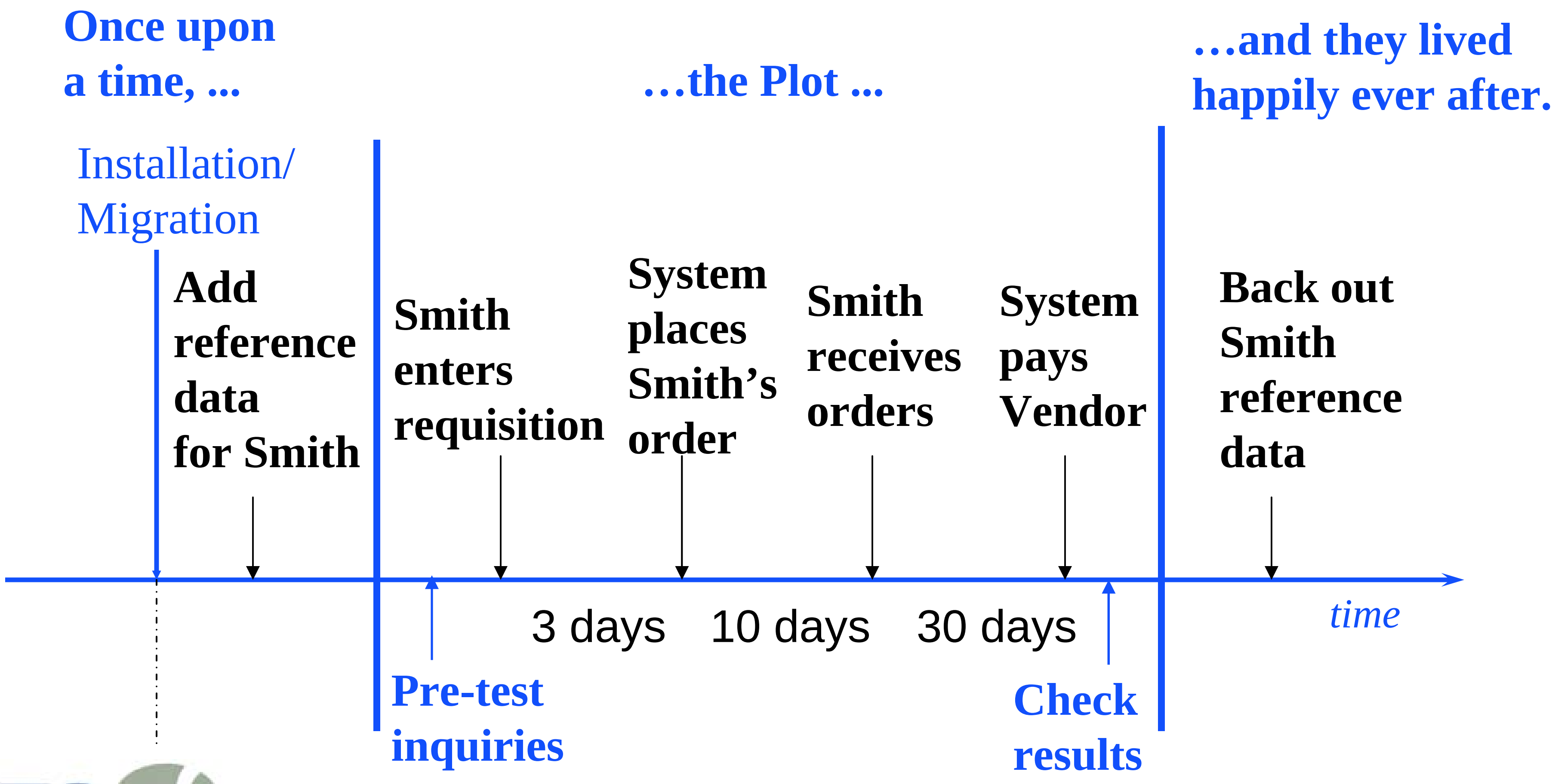
Pre-test inquiries

Check results

time

2. How do you design System Tests? (continued)

The Story of the Acquisition Test



Once upon a time, ...

…the Plot ...

…and they lived happily ever after.

Installation/ Migration

Add reference data for Smith

Smith enters requisition

System places Smith's order

Smith receives orders

System pays Vendor

Back out Smith reference data

Pre-test inquiries

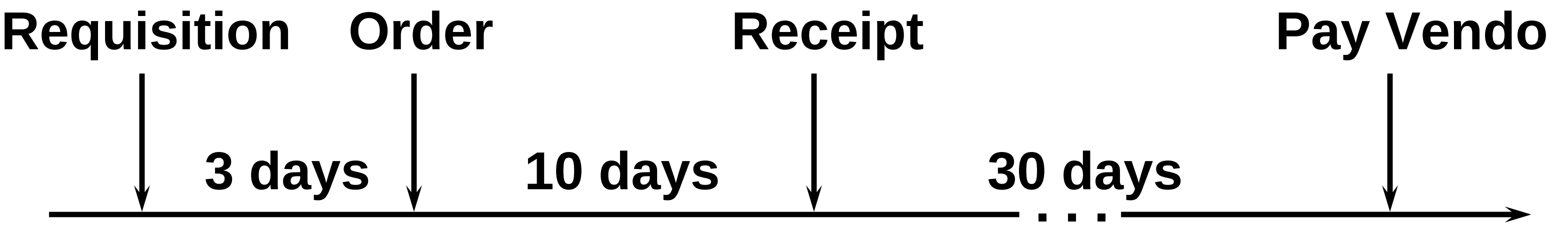
Check results

3. How do you make a test repeatable?

- Why is repeatability important in System Testing?
- Obstacles to repeatability
  - Remnants Problem
  - Common Sandbox Problem
  - Self-Competition Problem

4. How do you solve cycle acceleration problems?

Typical Chronology for Acquisition Process Business Flow



Requisition

Order

Receipt

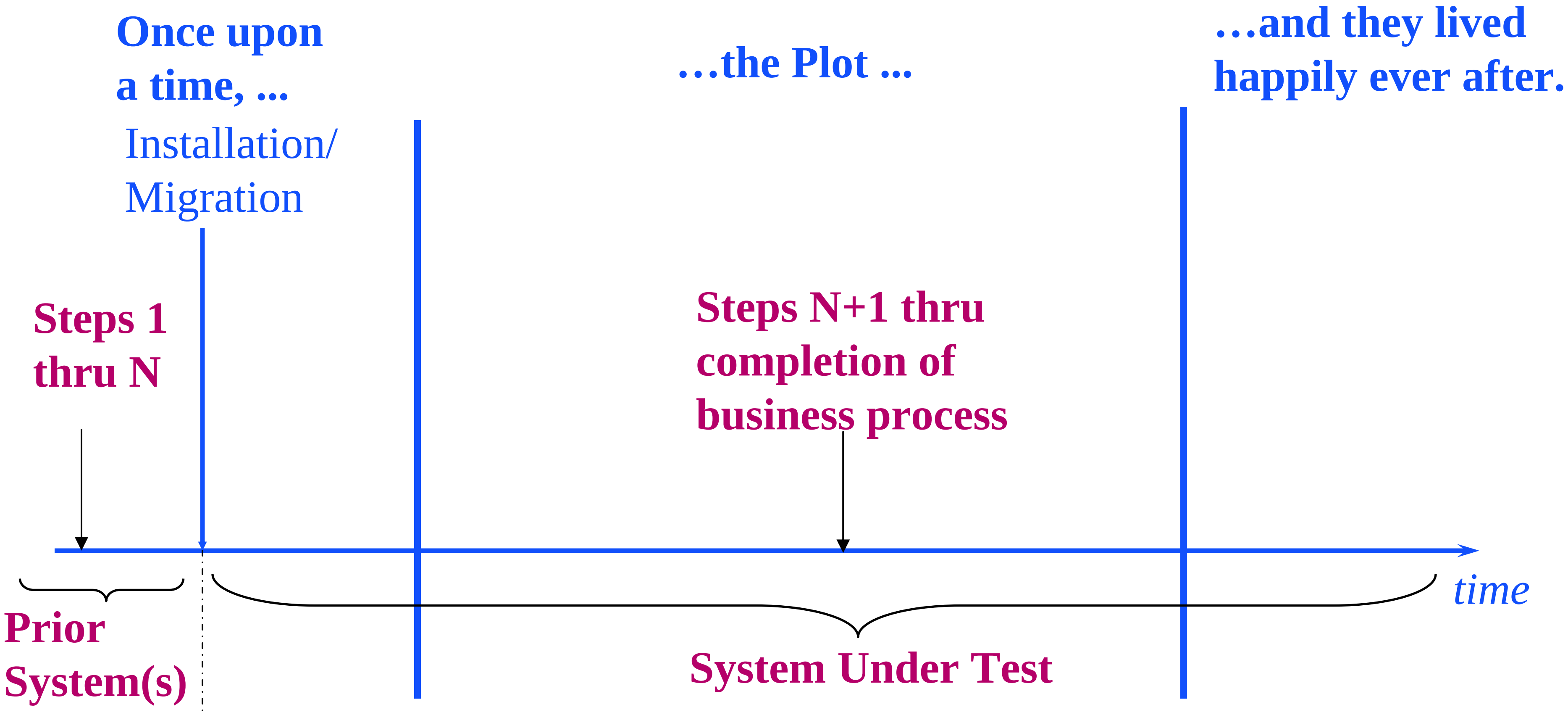
Pay Vendor

3 days

10 days

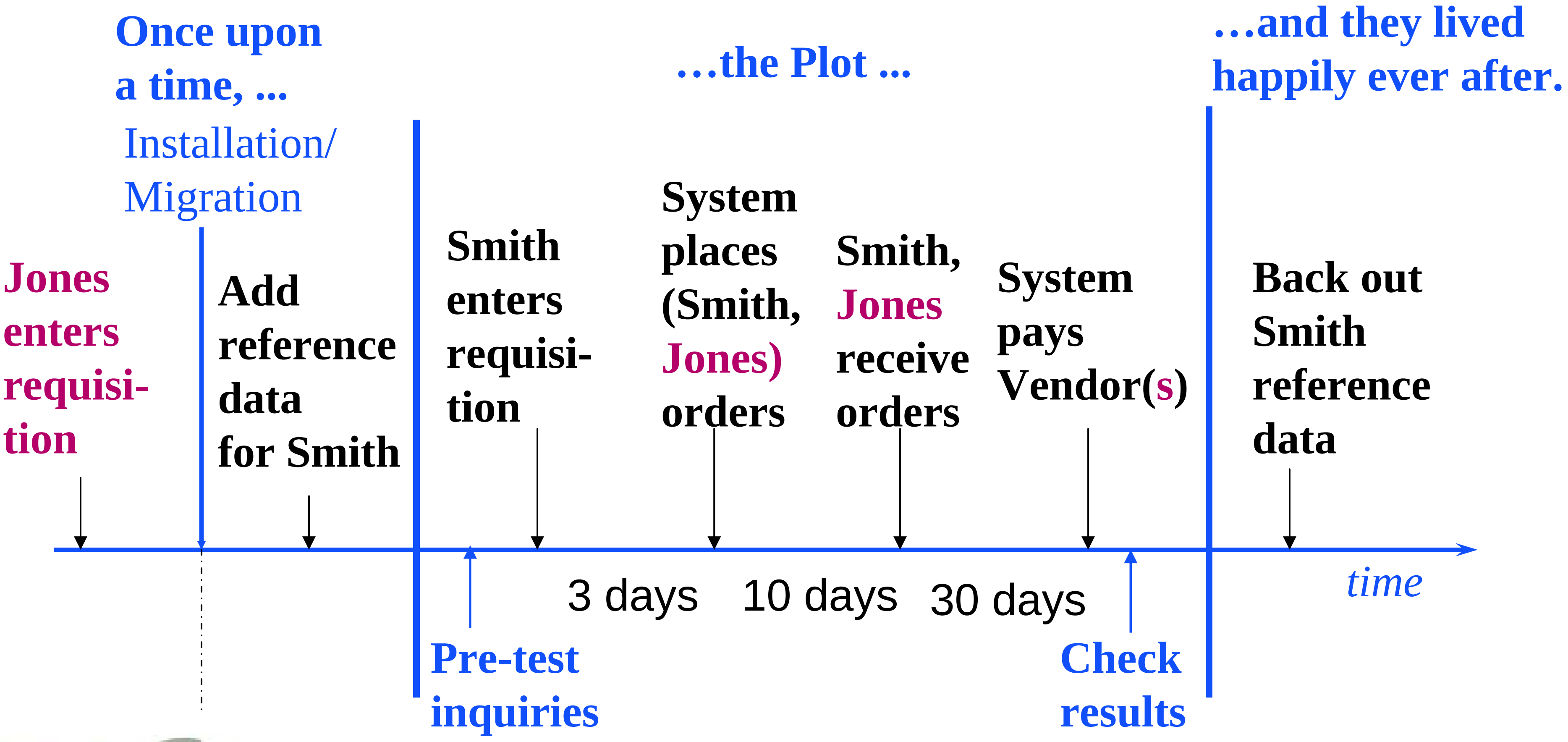
30 days

5. How do you test transactions-in-progress?



5. How do you test transactions-in-progress? (cont.)

The Story of the Acquisition Test



Once upon a time, ...

Installation/ Migration

Jones enters requisition

Add reference data for Smith

…the Plot ...

Smith enters requisition

System places (Smith, Jones) orders

Smith, Jones receive orders

System pays Vendor(s)

3 days

10 days

30 days

Pre-test inquiries

Check results

…and they lived happily ever after.

Back out Smith reference data

time

6. How do you leverage an existing test to cover additional data variations?

Two ways to add variations:

1. Parameterize (sequential approach)
2. Add new situations/characters/plot lines to the test story (parallel approach)

How does the tester determine which approach to apply?

7. What should you measure?

$$\text{System Test Team Effectiveness} = \frac{\text{Defects Avoided X 100}}{\text{Defects Avoided + Defects Missed}}$$

The Formula for Success in System Testing

System Testing Success =

(Technical Excellence)

+

(Nurturing Front-Loaded Quality)

where:

(Technical Excellence) >0

(Nurturing Front-Loaded Quality) >0