SoftServe | Empowering your Business through Software Development

Test Planning Seminar

Part #1

"Whatever begun with planning, ends in a victory"

Agenda

- **Part 1:**
  - Introduction
  - SoftServe template
  - Overview of Major Sections. Examples
- **Part 2:**
  - Test Planning Activities:
    - Test Scoping
    - Test Approach development
    - Estimating
    - Scheduling
    - Execution and Monitoring

SoftServe

Empowering your Business through Software Development

Introduction

Introduction: Basic Terms

- **Testing** — engineering process of investigation conducted to provide stakeholders with information about the quality of the product or service under test.
- **Test Plan** — a document detailing a systematic approach to testing a system such as a machine or software. The main artifact of test planning.
- **Test Planning** — activity aimed at determining test objectives and scope, approach, resources, environment, as well as estimates and schedule of project testing activities.

Test Planning: 5 Ws

Who?	• Testing roles and responsibilities
What?	• Test scope – items, features to be tested, features not to be tested
Where?	• Product operational environment and configurations
When?	• Schedule and testing workflow
Why?	• To mitigate project stakeholders risks related to product or service quality
How?	• Test strategy and approach

Test Planning: Myths

- **Myth #1**

  Test Plan is the set of test cases

- **Myth #2**

  Both Test Planning as activity and Test Plan do not exist in Agile

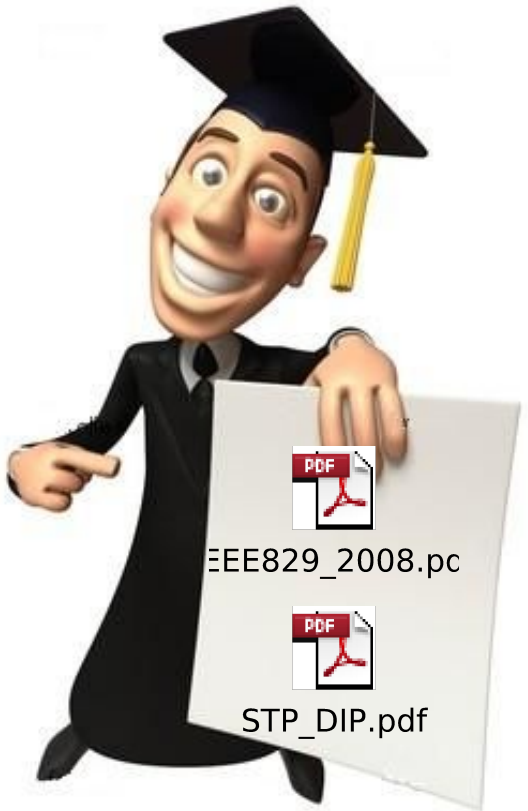
Standards Regulating Test Plan

- Institute of Electrical and Electronics Engineers

  IEEE 829. Standard for Test Documentation

- US Department of Defense's standard

  MIL-STD-498 (Military-Standard-498)



Master Test Plan

1. Introduction
- 1.1. Document identifier
- 1.2. Scope
- 1.3. References
- 1.4. System overview and key features
- 1.5. Test overview

2. Details of the Master Test Plan
- 2.1. Test processes including definition of test levels
- 2.2. Test documentation requirements
- 2.3. Test administration requirements
- 2.4. Test reporting requirements

3. General
- 3.1. Glossary
- 3.2. Document change procedures and history

Test Level Plan

1. Introduction

1.1. Document identifier

1.2. Scope

1.3. References

1.4. Level in the overall sequence

1.5. Test classes and overall test conditions

2. Details for this level of test plan

2.1 Test items and their identifiers

2.2 Test Traceability Matrix

2.3 Features to be tested

2.4 Features not to be tested

2.5 Approach

2.6 Item pass/fail criteria

2.7 Suspension criteria and resumption requirements

2.8 Test deliverables

3. Test management

3.1 Planned activities and tasks; test progression

3.2 Environment/infrastructure

3.3 Responsibilities and authority

3.4 Interfaces among the parties involved

3.5 Resources and their allocation

3.6 Training

3.7 Schedules, estimates, and costs

3.8 Risk(s) and contingency(s)

4. General

4.1 Quality assurance procedures

4.2 Metrics

4.3 Test coverage

4.4 Glossary

4.5 Document change procedures and history

SoftServe | Empowering your Business through Software Development

SoftServe Template

SoftServe Template

Test Plan

1 Introduction
- 1.1 Purpose
- 1.2 Overview
- 1.3 References
- 1.4 Definitions, Acronyms and Abbreviations

2 Test Items
- 2.1 Features To Be Tested
- 2.2 Features Not To Be Tested

3 Test Strategy
- 3.1 Test Approach
  - 3.1.1 Test Focus
  - 3.1.2 Test Coverage
  - 3.1.3 Test Types
- 3.2 QA Efforts Workflow
- 3.3 Test Design
- 3.4 Test Execution
  - 3.4.1 Test Status Reporting
  - 3.4.2 Defect Tracking
- 3.5 Test Deliverables
- 3.6 Test Environment
  - 3.6.1 Configurations
  - 3.6.2 Testing Tools
- 3.7 Entrance Criteria
- 3.8 Pass/Fail Criteria
- 3.9 Exit Criteria

4 Roles and Responsibilities

5 Schedule

Agile Test Plan

1 Introduction
- 1.1 Purpose
- 1.2 Overview
- 1.3 References
- 1.4 Definitions, Acronyms and Abbreviations

2 Test Items
- 2.1 Features To Be Tested
- 2.2 Features Not To Be Tested

3 Test Strategy
- 3.1 Test Approach
  - 3.1.1 Zero Sprint
  - 3.1.2 Release Sprint
  - 3.1.3 Test Focus
  - 3.1.4 Test Coverage of Release
  - 3.1.5 Test Types
- 3.2 QC Efforts Workflow
- 3.3 Test Design
- 3.4 Test Execution
  - 3.4.1 Test Status Reporting
  - 3.4.2 Defect Tracking
- 3.5 Test Deliverables
- 3.6 Test Environment
  - 3.6.1 Configurations
  - 3.6.2 Testing Tools
- 3.7 Entrance Criteria
- 3.8 Pass/Fail Criteria
- 3.9 Acceptance Criteria

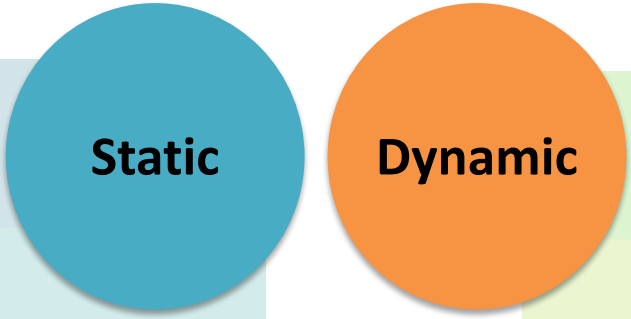
SoftServe Templates

Based on IEEE 829

SoftServe | Empowering your Business through Software Development

Major Sections of a Test Plan

Major Parts: Static vs. Dynamic



- Introduction
- Test Items
- Item Pass/Fail Criteria
- Staffing and Training Needs
- and other…

- Features to Be Tested
- Features Not to Be Tested
- Schedule
- Risks and Contingencies
- and other…

Major Sections: Introduction

Identifier

– Name which uniquely identifies the test plan document

<ClientName>_<ProjectName>_<version>.<document_format>

Introduction

- Purpose of the project
- System under test overview (both from technical and business standpoint)
- Project references properly versioned
- Definitions, Acronyms and Abbreviations

Major Sections: Introduction

- **1. Introduction**

  1.1 Purpose

- The purpose of this document is to work out in details the QC activities required to be performed for SystemX project; to define the test strategy and approach to testing; to define the scope of the QC activities and to identify the responsibilities.
- **1.2 Overview**
- SystemX is a product that allows users to organize and manage Business Cards, Receipts, Expense Reports, General Documents and generate Tax Information. Its primary purpose is to allow users to organize the paper clutter that has become a part of so many work environments. It is sold as a retail product to individuals and small businesses. The product is sold in both Brick and Mortar as well as Web based retail stores. The software is always sold in combination with a small portable scanner.

Example

Major Sections: Test Items

Test Items

- Test plan section describing the test objects which are the subject of the test plan
  - Components that constitute the test object
  - Interfacing components required for test object testing
  - Procedures and instructions
  - Documents

Test Items: Let's Practice!

System under test highlights:

Client-server application with web interface that facilitates online interviews management and execution processes

- **Test Items**

  The following components are planned to be tested:

  1. <Client>'s web-site
  2. Interview Manager sub system
  3. Users management sub system
  4. Accounts management sub system and logout

Major Sections: Features

Features to be Tested

List of features which will be tested (defined for each of the test items)

Features Not to be Tested

List of features which will NOT be tested (defined for each of the test items)

Major Sections: Quality Attributes

Features To Be Tested: Let's Practice!

- Features to Be Tested

System under test highlights:

Client-server application with web interface that facilitates online interviews management and execution processes.

2. Interview Manager sub system
    1. Interviews Manager home page
    2. Managing calendar, recently recorded, pending interviews; Creating new interviews
3. Users management sub system
    1. Adding/editing/viewing users (preferences)
    2. Managing user roles
    3. Searching for users
4. Accounts management sub system and logout
    1. Adding/editing/deleting accounts
    2. Signing in as another user
    3. Searching for accounts; log out

Major Sections: Strategy/ Approach

Test Strategy/Approach

- The MOST important part of a test plan, describing the implementation of testing activities to meet quality expectation goals

Includes the following:

- Test levels, types, techniques and methods to be used
- Sequence and description of testing activities
- Distribution of resources

Test Strategy is an **outline** that describes the **testing portion** of the software development cycle

Separate document usually

Test Approach describes the **implementation** of the **test strategy** for a specific project

Part of a Test Plan

Major Sections: Entry/Exit Criteria

Entry/Exit Criteria

- **Entry** – conditions or process that must be present before a process can begin
- **Exit** – conditions or process that must be present before a cycle completes

Major Sections: Entry/Exit Criteria

- **Entry Criteria**
  1. Test Plan is reviewed and approved by <Client>.
  2. Test environments (staging, production sites) are available.
  3. Test data is prepared.
  4. Tools, defined in the "Tools" section of this document, are configured and available to QC Team. Access credentials are provided.
- **Exit Criteria**
  1. All testing tasks are completed.
  2. Deliverables are prepared and placed on Portal.

Example

Major Sections: Suspension & Resumption

Suspension Criteria and Resumption Requirements

- Section describes conditions that determine if a testing object should be returned back to development team (*testing stopped*) and requirements for taking back test object for testing.

Major Sections: Suspension & Resumption

Suspension Criteria and Resumption Requirements

Functional testing will be stopped in case:

1. Number of Crash - Data Loss defects is constantly growing and reaches five items per single testing cycle.

Following criteria must be met in order to proceed further with functional testing:

- ✔ Crash - Data Loss defects are fixed and verified as passed
- ✔ Automated tests are written for Crash-Data Loss defects and are included into Continuous build integration and verification procedures.
- ✔ 100% unit tests passed
- ✔ 100% smoke test passed

Example

Major Sections: Pass/Fail Criteria

Item Pass/Fail Criteria

- These are measurable definitions of criteria clearly identifying that a *test object* can be *released* from testing

Should not be mixed up with System or User Acceptance criteria

Major Sections: Pass/Fail Criteria

Item Pass/Fail Criteria

1. Functional testing is not less than 98% passed on Test Environment with no P1 and P2 functional defects remain non-addressed.
2. Performance testing is 100% passed on Staging environment.
3. All Usability defects are addressed.

Example

Major Sections: Deliverables

Test Deliverables

- Section describing the work products produced during test activities.

  Should include the testing work product's attributes:

  - name,
  - description,
  - format and version,
  - any standards work product should correspond to.

- Targets documents, tools and data, etc.

Major Sections: Tasks and Environment

Testing Tasks

- Set of tasks necessary to prepare for and perform testing, accompanied with entry/exit criteria and responsible person

May be tracked within management tool

Environmental Needs

- Specifies both the necessary and the desired properties of the test environment and any relevant test data

Should include:

- hardware and software
- test support tools and databases
- support personnel
- security level
- test level specifications requirements

Major Sections: Staff & Responsibilities

Responsibilities

- Mapping of project team organizational roles to testing tasks

Staffing and Training Needs

- Set of required human resources' skills mapped to testing tasks
- Test and domain-specific knowledge per each role
- Trainings required and how they are going to be carried out

Major Sections: Schedule

Schedule

- Commonly this is a dynamic set of milestones for testing team

Milestones are defined basing on following information:

- test effort estimation
- resources availability
- project development lifecycle
- risks and contingencies

Part 2

Major Sections: Risks

Risks and Contingencies

- List of possible events that may prevent testing objectives/goals from being achieved; or being achieved not in time.

Includes information per each risk identified:

- Rank
  - Probability x Impact
- Response
  - Accept, Avoid, Mitigate, Transfer
- Mitigation/Contingency Plan
  - Actions you take in case risk happens

Risk-Based Testing

- **Risk-based testing**
  - is a highly effective testing strategy that can be used to find and fix the most important problems as quickly as possible.

  (remember Pareto principle?)

- **How to use**
  1. Analyze the impact and probability of failure of the functionality or other quality characteristics being tested
  2. Basing on analysis, adjust the tests and the extent of testing according to the risk priority levels
  3. As new risk appears, re-plan

Major Sections: Approvals

Approvals

- List of project stakeholders with their roles identified who need to sign off a test plan as proof of notice.

Should include, but not limited to:

- **Project manager** – recognition of budget approval
- **Development leader** – confirmation of test approach and involvement of dev. team resources
- **Customer representative** – confirmation of proposed testing procedures
- **Test team** – "act of commitment"☺️

Test Plan Example

Questions are Welcome! ☺️

SoftServe | Empowering your Business through Software Development

Test Planning Seminar

Part #2

Agenda

- **Part 1:**
  - Introduction
  - SoftServe template
  - Overview of Major Sections. Examples

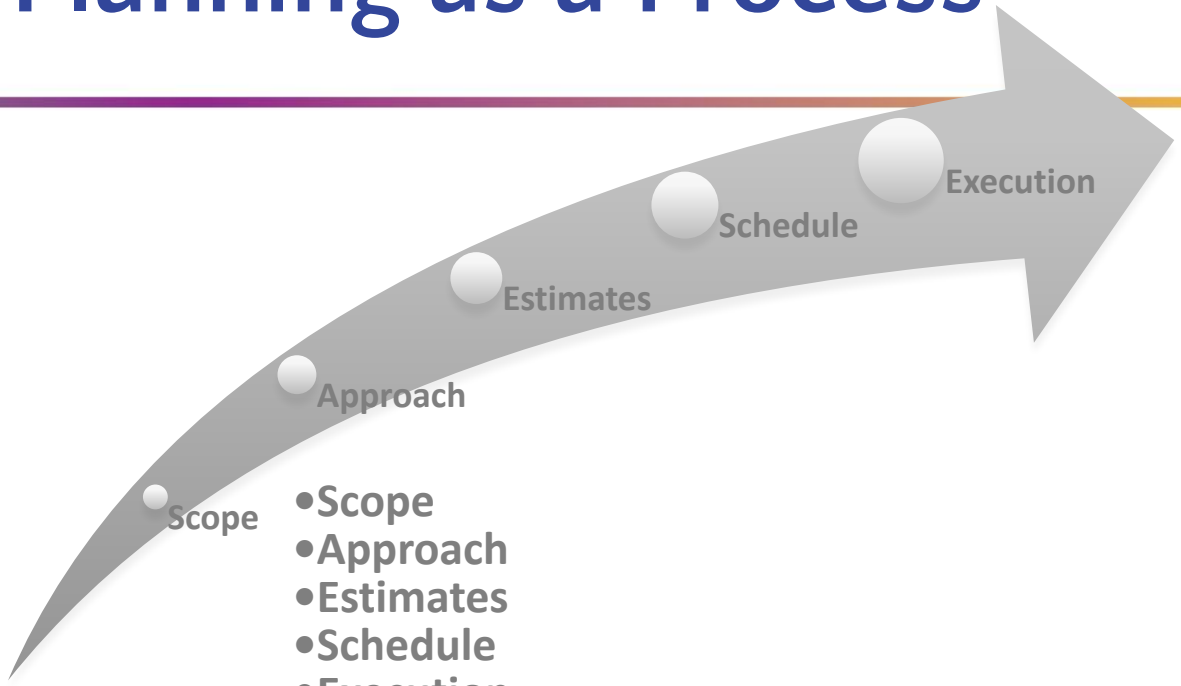
- **Part 2:**
  - Test Planning Activities:
    - Test Scoping
    - Test Approach development
    - Estimating
    - Scheduling
    - Execution and Monitoring

SoftServe | Empowering your Business through Software Development

Test Planning Activities:
Test Scope

Scope – Approach– Estimate – Schedule – Execution &Monitoring

Test Planning as a Process



- Scope
- Approach
- Estimates
- Schedule
- Execution

- **Scope**
- **Approach**
- **Estimates**
- **Schedule**
- **Execution**


Test Scope
=
Test Items
+
Test Items' Quality Characteristics

Test Scope

- **Test Item** – an object that can be architecturally, functionally or procedurally separated out of the system

- **Test Item's quality characteristics** can be assessed by:
  - **ISO/IEC 9126** – an international standard for the evaluation of software quality
  - **FURPS** – model for classifying software quality attributes, both functional & non-functional requirements, developed at Hewlett-Packard

ISO/IEC 9126. Quality Model

- **Functionality**
  - Suitability
  - Accuracy
  - Interoperability
  - Security
  - Functionality Compliance
- **Reliability**
  - Maturity
  - Fault Tolerance
  - Recoverability
- **Usability**
  - Understandability
  - Learnability
  - Operability
  - Attractiveness
- **Maintainability**
  - Analyzability
  - Changeability
  - Stability
  - Testability
- **Efficiency**
  - Time Behavior
  - Resource Utilization
- **Portability**
  - Adaptability
  - Installability
  - Co-Existence
  - Replaceability

FURPS Quality Model

- **Functionality**
  - Feature set
  - Capabilities
  - Generality
  - Security
- **Usability**
  - Human factors
  - Aesthetics
  - Consistency
  - Documentation
- **Reliability**
  - Frequency/severity of failure
  - Recoverability
  - Predictability
  - Accuracy
  - Mean time to failure
- **Performance**
  - Speed
  - Efficiency
  - Resource consumption
  - Throughput
  - Response time
- **Supportability**
  - Testability
  - Extensibility
  - Adaptability
  - Maintainability
  - Compatibility
  - Configurability
  - Serviceability
  - Installability
  - Localizability
  - Portability

Practical Guide to Test Scoping

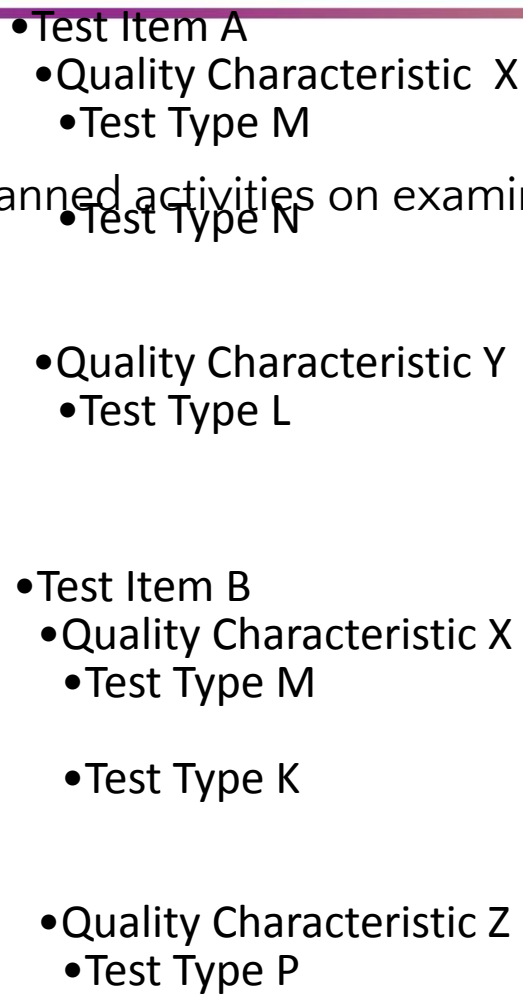
1. Identify and write down your Test Items.
2. Evaluate each Test Item against set of quality attributes or characteristics and note down applicable ones.
3. Ask major project stakeholders (customer, development lead, architect) to prioritize identified quality attributes/ characteristics per each Test Item.
4. Identify objectives/goals (might be measured by item pass/fail criteria) for your test activities basing on prioritized quality attributes list and responsibilities (internal quality characteristics are usually assessed by development team)
5. Now you are ready to the next step – Test Approach development.

SoftServe | Empowering your Business through Software Development

Test Planning Activities:
Test Approach

Test Approach in 3 Steps

- **Test Approach**
  - Implementation of planned activities on examining internal and external quality of Test Items.



Practical Guide to Test Approach

- Examination Method
  - Dynamic
  - Static
- Testing types to be applied
  - If testing type is applicable for several Test Items, you need to specify any differences during execution.
- Execution method
  - Manual, Automated, Semi-Automated
- Development and deployment process
  - New changes integration principle (regression testing and impact analysis)
- Environment Requirements

Test Levels. Need to Know

- While developing Test Approach think at which **test level** you are going to examine the system:
  - Component
  - Integration
  - System
  - Acceptance (System Acceptance)

Let's Practice!

1. Identify test items
2. Select applicable quality characteristics
3. Propose test approach

SoftServe | Empowering your Business through Software Development

Test Planning Activities:
Estimating

Estimating

- **Estimation** is the calculated approximation of a result which is usable even if input data may be incomplete or uncertain.

  Your Result – **man/hours** needed to fulfill testing tasks (costs are not your responsibility at this moment)

The most useful estimation approaches:

Analogy, Expert Judgment, Disaggregation

Practical Guide to Tests Estimating

1. Write down all the tasks needed to fulfill your Test Approach.
2. Disaggregate tasks to “estimatable” items.
3. Using analogy or previous experience give your values for time to each peace in man/hours (mins, months, etc.) for Optimistic, Pessimistic, Most Likely estimates.
4. Using **PERT** calculate your Expected Time

$$T_E = (O + 4ML + P)/6$$

5. Calculate your Totals.
6. Ask several experts you trust to do step 3 (1-2 optionally) and revise final numbers.

SoftServe | Empowering your Business through Software Development

Test Planning Activities:
Scheduling

Scheduling

- **Scheduling** is the process of deciding how to commit resources between a variety of possible tasks.
  - Start/end time to specify
  - Creating sequence of events

Practical Guide to Scheduling

- As test scheduling always depends on multi-factors analysis, consider the following:
  - Sequence driven order
    - Test infrastructure set up first
    - Preparation activities second (requirements analysis, test data and tests design)
    - Execution activities third
  - Business driven order
    - Highest priority features first; milestones to meet
  - Resources availability driven order
    - Human resources
    - Tools and environment, time and cost
  - Development process driven order
    - New changes releasing and integration (regression testing, stabilization, code freeze slots to consider)

SoftServe | Empowering your Business through Software Development

Test Planning Activities:
Execution & Monitoring

Test Approach Execution

- **Test Execution** - a step by step implementation of selected approach to meet the objectives set.
- Except of actual testing tasks execution following means of test planning to be considered:
  - **Initiating Testing Tasks**
    - Evaluating entry criteria
  - **Monitoring** (measurement and metrics) and **Reporting**
    - Product quality monitoring
    - Testing processes quality monitoring
    - Schedule monitoring
  - **Reacting to Changes** (re-planning)
  - **Evaluating Test Completion**
    - Have we tested enough?

Monitoring in Metrics

- **Test Process Metrics**
  - **Test Design**
    - number of Features to be Tested covered by Tests vs. Total number of Features to be Tested (test coverage)
    - number of Tests Planned vs. Ready for Execution
    - total time spent on Preparation vs. Estimated time
  - **Test Execution and Progress**
    - number of Tests Executed vs. Tests Planned
    - number of Tests Passed, Failed and Blocked
    - total number of Tests Passed by Functional Areas/Feature to be Tested
    - total number of defects Verified Failed vs. total number of defects Closed
    - Total Time Spent on Execution vs. Estimated Time
    - Average response time to defect

Example

Monitoring in Metrics

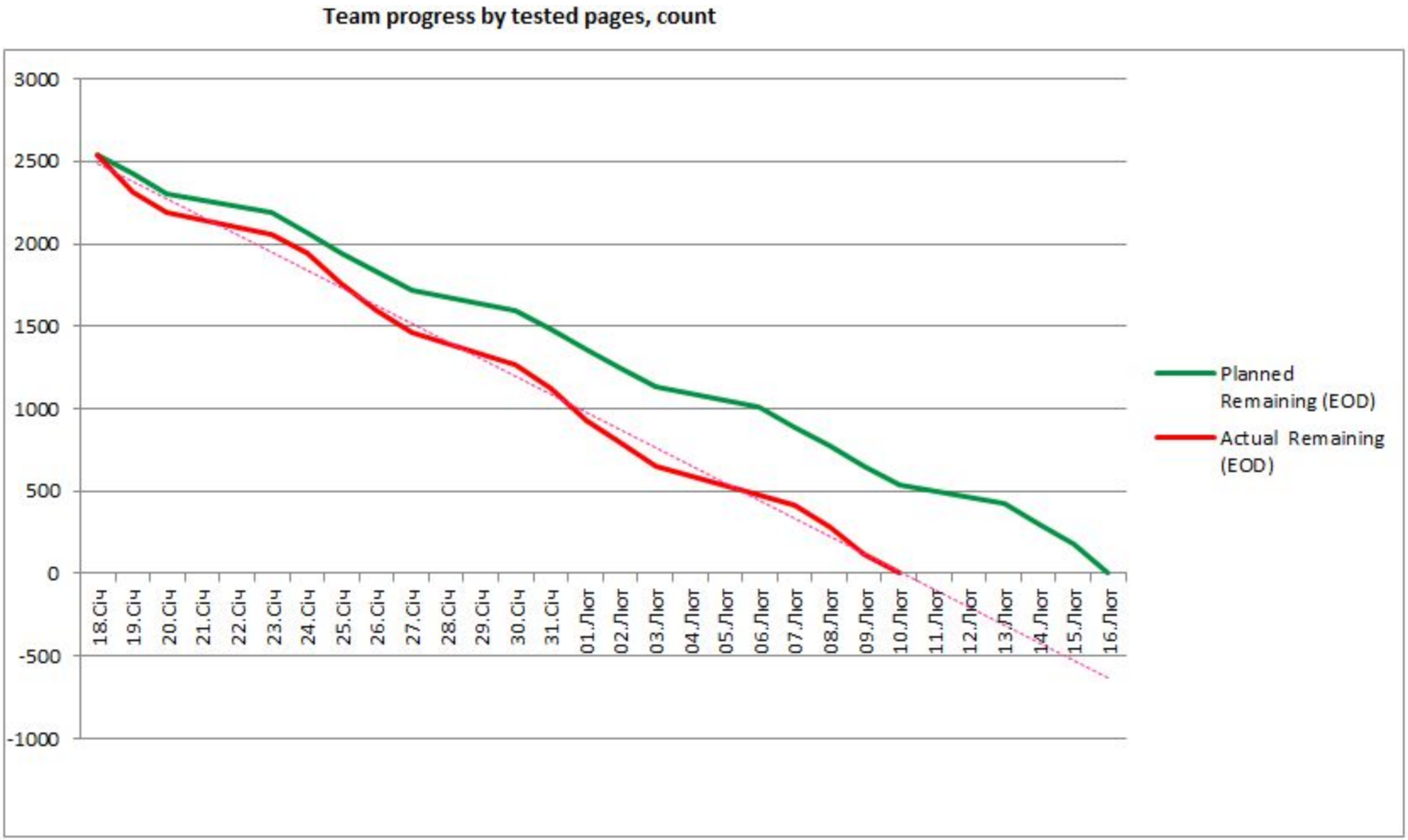
Test Product Metrics

– Defects Based Analysis

- defects arrival
- total number of defects Opened and Closed per Period
- defects distribution totals by Severity per Period
- defects distribution totals by Functional Areas/Feature to be Tested by Severity/Testing Type/Priority per Period
- CRUD – customer reported unique defects (bugs) rate (total number of customer reported defects vs. total defects identified by test team)

Example

Monitoring in Metrics



Reporting

- Status reporting/Test Summary Report
  - **Agreed Format**
    - Document, e-mail, status call, informal reporting
  - **Timelines**
    - Daily, weekly, monthly, per milestones
  - **Typical scope**
    - Accomplishments achieved since last reporting
    - Planned goals to achieve till next reporting
    - Statistics (metrics on progress and product quality)
    - Issues

Test Summary Report -Document detailing results of testing, conclusions and recommendations made

Questions are Welcome! ☺️


Thank You!
Good Luck in Planning
to succeed!