

1. Record Nr.	UNINA9910220533803321
Autore	Hoed P. den
Titolo	Planning als onderneming / / P. den Hoed, W. G. M. Salet, H. van der Sluijs
Pubbl/distr/stampa	's-Gravenhage : , : WRR, , 1983
Descrizione fisica	1 online resource (160 pages)
Collana	Voorstudies en achtergronden ; ; Volume 34
Disciplina	350
Soggetti	Political planning Public administration
Lingua di pubblicazione	Olandese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Sommario/riassunto	Schets van de historische rol van planning bij de overheid en de wijze, waarop planning in de toekomst zou moeten functioneren NL-ZmNBD.

2. Record Nr.	UNINA9910830133203321
Autore	Homes Bernard
Titolo	Advanced testing of systems-of-systems 2 : practical aspects / / Bernard Homes
Pubbl/distr/stampa	John Wiley & Sons, Inc London, England ; ; Hoboken, New Jersey : , : ISTE Ltd : , : John Wiley & Sons, Inc., , [2022] ©2022
ISBN	1-394-18848-X 1-394-18846-3
Descrizione fisica	1 online resource (306 pages)
Collana	Computer engineering series
Disciplina	929.605
Soggetti	Computer software
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Cover -- Title Page -- Copyright Page -- Contents -- Title Page -- Copyright Page -- Contents -- Dedication and Acknowledgments -- Preface -- Chapter 1. Test Project Management -- 1.1. General principles -- 1.1.1. Quality of requirements -- 1.1.2. Completeness of deliveries -- 1.1.3. Availability of test environments -- 1.1.4. Availability of test data -- 1.1.5. Compliance of deliveries and schedules -- 1.1.6. Coordinating and setting up environments -- 1.1.7. Validation of prerequisites - Test Readiness Review (TRR) -- 1.1.8. Delivery of datasets (TDS) -- 1.1.9. Go-NoGo decision - Test Review Board (TRB) -- 1.1.10. Continuous delivery and deployment -- 1.2. Tracking test projects -- 1.3. Risks and systems-of-systems -- 1.4. Particularities related to SoS -- 1.5. Particularities related to SoS methodologies -- 1.5.1. Components definition -- 1.5.2. Testing and quality assurance activities -- 1.6. Particularities related to teams -- Chapter 2. Testing Process -- 2.1. Organization -- 2.2. Planning -- 2.2.1. Project WBS and planning -- 2.3. Control of test activities -- 2.4. Analyze -- 2.5. Design -- 2.6. Implementation -- 2.7. Test execution -- 2.8. Evaluation -- 2.9. Reporting -- 2.10. Closure -- 2.11. Infrastructure management -- 2.12. Reviews -- 2.13. Adapting processes -- 2.14. RACI matrix -- 2.15. Automation of processes or

tests -- 2.15.1. Automate or industrialize? -- 2.15.2. What to automate? -- 2.15.3. Selecting what to automate -- Chapter 3. Continuous Process Improvement -- 3.1. Modeling improvements -- 3.1.1. PDCA and IDEAL -- 3.1.2. CTP -- 3.1.3. SMART -- 3.2. Why and how to improve? -- 3.3. Improvement methods -- 3.3.1. External/internal referential -- 3.4. Process quality -- 3.4.1. Fault seeding -- 3.4.2. Statistics -- 3.4.3. A posteriori -- 3.4.4. Avoiding introduction of defects -- 3.5. Effectiveness of improvement activities. 3.6. Recommendations -- Chapter 4. Test, QA or IV&V Teams -- 4.1. Need for a test team -- 4.2. Characteristics of a good test team -- 4.3. Ideal test team profile -- 4.4. Team evaluation -- 4.4.1. Skills assessment table -- 4.4.2. Composition -- 4.4.3. Select, hire and retain -- 4.5. Test manager -- 4.5.1. Lead or direct? -- 4.5.2. Evaluate and measure -- 4.5.3. Recurring questions for test managers -- 4.6. Test analyst -- 4.7. Technical test analyst -- 4.8. Test automator -- 4.9. Test technician -- 4.10. Choose our testers -- 4.11. Training, certification or experience? -- 4.12. Hire or subcontract) -- 4.12.1. Effective subcontracting -- 4.13. Organization of multi-level test teams -- 4.13.1. Compliance, strategy and organization -- 4.13.2. Unit test teams (UT/CT) -- 4.13.3. Integration testing team (IT) -- 4.13.4. System test team (SYST) -- 4.13.5. Acceptance testing team (UAT) -- 4.13.6. Technical test teams (TT) -- 4.14. Insourcing and outsourcing challenges -- 4.14.1. Internalization and collocation -- 4.14.2. Near outsourcing -- 4.14.3. Geographically distant outsourcing -- Chapter 5. Test Workload Estimation -- 5.1. Difficulty to estimate workload -- 5.2. Evaluation techniques -- 5.2.1. Experience-based estimation -- 5.2.2. Based on function points or TPA -- 5.2.3. Requirements scope creep -- 5.2.4. Estimations based on historical data -- 5.2.5. WBS or TBS -- 5.2.6. Agility, estimation and velocity -- 5.2.7. Retroplanning -- 5.2.8. Ratio between developers - testers -- 5.2.9. Elements influencing the estimate -- 5.3. Test workload overview -- 5.3.1. Workload assessment verification and validation -- 5.3.2. Some values -- 5.4. Understanding the test workload -- 5.4.1. Component coverage -- 5.4.2. Feature coverage -- 5.4.3. Technical coverage -- 5.4.4. Test campaign preparation -- 5.4.5. Running test campaigns -- 5.4.6. Defects management. 5.5. Defending our test workload estimate -- 5.6. Multi-tasking and crunch -- 5.7. Adapting and tracking the test workload -- Chapter 6. Metrics, KPI and Measurements -- 6.1. Selecting metrics -- 6.2. Metrics precision -- 6.2.1. Special case of the cost of defaults -- 6.2.2. Special case of defects -- 6.2.3. Accuracy or order of magnitude? -- 6.2.4. Measurement frequency -- 6.2.5. Using metrics -- 6.2.6. Continuous improvement of metrics -- 6.3. Product metrics -- 6.3.1. FTR: first time right -- 6.3.2. Coverage rate -- 6.3.3. Code churn -- 6.4. Process metrics -- 6.4.1. Effectiveness metrics -- 6.4.2. Efficiency metrics -- 6.5. Definition of metrics -- 6.5.1. Quality model metrics -- 6.6. Validation of metrics and measures -- 6.6.1. Baseline -- 6.6.2. Historical data -- 6.6.3. Periodic improvements -- 6.7. Measurement reporting -- 6.7.1. Internal test reporting -- 6.7.2. Reporting to the development team -- 6.7.3. Reporting to the management -- 6.7.4. Reporting to the clients or product owners -- 6.7.5. Reporting to the direction and upper management -- Chapter 7. Requirements Management -- 7.1. Requirements documents -- 7.2. Qualities of requirements -- 7.3. Good practices in requirements management -- 7.3.1. Elicitation -- 7.3.2. Analysis -- 7.3.3. Specifications -- 7.3.4. Approval and validation -- 7.3.5. Requirements management -- 7.3.6. Requirements and business knowledge management -- 7.3.7. Requirements and project management -- 7.4. Levels of requirements

-- 7.5. Completeness of requirements -- 7.5.1. Management of TBDs and TBCs -- 7.5.2. Avoiding incompleteness -- 7.6. Requirements and agility -- 7.7. Requirements issues -- Chapter 8. Defects Management -- 8.1. Defect management, MOA and MOE -- 8.1.1. What is a defect? -- 8.1.2. Defects and MOA -- 8.1.3. Defects and MOE -- 8.2. Defect management workflow -- 8.2.1. Example -- 8.2.2. Simplify. 8.3. Triage meetings -- 8.3.1. Priority and severity of defects -- 8.3.2. Defect detection -- 8.3.3. Correction and urgency -- 8.3.4. Compliance with processes -- 8.4. Specificities of TDDs, ATDDs and BDDs -- 8.4.1. TDD: test-driven development -- 8.4.2. ATDD and BDD -- 8.5. Defects reporting -- 8.5.1. Defects backlog management -- 8.6. Other useful reporting -- 8.7. Don't forget minor defects -- Chapter 9. Configuration Management -- 9.1. Why manage configuration? -- 9.2. Impact of configuration management -- 9.3. Components -- 9.4. Processes -- 9.5. Organization and standards -- 9.6. Baseline or stages, branches and merges -- 9.6.1. Stages -- 9.6.2. Branches -- 9.6.3. Merge -- 9.7. Change control board (CCB) -- 9.8. Delivery frequencies -- 9.9. Modularity -- 9.10. Version management -- 9.11. Delivery management -- 9.11.1. Preparing for delivery -- 9.11.2. Delivery validation -- 9.12. Configuration management and deployments -- Chapter 10. Test Tools and Test Automation -- 10.1. Objectives of test automation -- 10.1.1. Find more defects -- 10.1.2. Automating dynamic tests -- 10.1.3. Find all regressions -- 10.1.4. Run test campaigns faster -- 10.2. Test tool challenges -- 10.2.1. Positioning test automation -- 10.2.2. Test process analysis -- 10.2.3. Test tool integration -- 10.2.4. Qualification of tools -- 10.2.5. Synchronizing test cases -- 10.2.6. Managing test data -- 10.2.7. Managing reporting (level of trust in test tools) -- 10.3. What to automate? -- 10.4. Test tooling -- 10.4.1. Selecting tools -- 10.4.2. Computing the return on investment (ROI) -- 10.4.3. Avoiding abandonment of tools and automation -- 10.5. Automated testing strategies -- 10.6. Test automation challenge for SoS -- 10.6.1. Mastering test automation -- 10.6.2. Preparing test automation -- 10.6.3. Defect injection/fault seeding. 10.7. Typology of test tools and their specific challenges -- 10.7.1. Static test tools versus dynamic test tools -- 10.7.2. Data-driven testing (DDT) -- 10.7.3. Keyword-driven testing (KDT) -- 10.7.4. Model-based testing (MBT) -- 10.8. Automated regression testing -- 10.8.1. Regression tests in builds -- 10.8.2. Regression tests when environments change -- 10.8.3. Prevalidation regression tests, sanity checks and smoke tests -- 10.8.4. What to automate? -- 10.8.5. Test frameworks -- 10.8.6. E2E test cases -- 10.8.7. Automated test case maintenance or not? -- 10.9. Reporting -- 10.9.1. Automated reporting for the test manager -- Chapter 11. Standards and Regulations -- 11.1. Definition of standards -- 11.2. Usefulness and interest -- 11.3. Implementation -- 11.4. Demonstration of compliance - IADT -- 11.5. Pseudo-standards and good practices -- 11.6. Adapting standards to needs -- 11.7. Standards and procedures -- 11.8. Internal and external coherence of standards -- Chapter 12. Case Study -- 12.1. Case study: improvement of an existing complex system -- 12.1.1. Context and organization -- 12.1.2. Risks, characteristics and business domains -- 12.1.3. Approach and environment -- 12.1.4. Resources, tools and personnel -- 12.1.5. Deliverables, reporting and documentation -- 12.1.6. Planning and progress -- 12.1.7. Logistics and campaigns -- 12.1.8. Test techniques -- 12.1.9. Conclusions and return on experience -- Chapter 13. Future Testing Challenges -- 13.1. Technical debt -- 13.1.1. Origin of the technical debt -- 13.1.2. Technical debt elements -- 13.1.3. Measuring technical debt -- 13.1.4.

Reducing technical debt -- 13.2. Systems-of-systems specific challenges -- 13.3. Correct project management -- 13.4. DevOps -- 13.4.1. DevOps ideals -- 13.4.2. DevOps-specific challenges -- 13.5. IoT (Internet of Things) -- 13.6. Big Data. 13.7. Services and microservices.

Sommario/riassunto

As a society today, we are so dependent on systems-of-systems that any malfunction has devastating consequences, both human and financial. Their technical design, functional complexity and numerous interfaces justify a significant investment in testing in order to limit anomalies and malfunctions. Based on more than 40 years of practice, this book goes beyond the simple testing of an application - already extensively covered by other authors - to focus on methodologies, techniques, continuous improvement processes, load estimates, metrics and reporting, which are illustrated by a case study. It also discusses several challenges for the near future. Pragmatic and clear, this book displays many examples and references that will help you improve the quality of your systems-of-systems efficiently and effectively and lead you to identify the impact of upstream decisions and their consequences. Advanced Testing of Systems-of-Systems 2 deals with the practical implementation and use of the techniques and methodologies proposed in the first volume.
