TechTarget | SearchSoftware Quality

E-toolkit

PRO+ Content

A Template for Software Requirements Gathering Techniques

Your toolkit for eliciting better techniques to optimize software development processes.

In this e-toolkit

In this e-toolkit:

Requirements gathering is an essential part of software development.

A strong software requirements specification (SRS) minimizes the time and effort required by developers to achieve desired goals and cut costs.

But with any people-centered development activity, a SRS can be difficult to implement, and mistakes may be hard to avoid.

This carefully- assembled toolkit, including the best methods for facilitating the requirements gathering process, will guide you as you tackle this important but challenging step.

Leverage this toolkit to help you make the most of software development processes, and avoid road bumps along the way.

In this e-toolkit

Prototyping

Prototypes range from the simple to the elaborate. But whether it's a paper sketch or an interactive digital design, a prototype should aid stakeholders and developers in anticipating requirements for a product.

- What is prototyping?
- Effective Prototyping for Software Makers -- Chapter 2, The Effective Prototyping Process: Prototyping is a process anyone can learn and master, according to author Jonathan Arnowitz. In this book excerpt, he describes the four phases of the prototyping process -- plan, specification, design and results -- and how to use prototyping to communicate software requirements, designs and ideas.
- Create screen prototypes for clear software requirements: A step by step approach to communicating with your customers and getting the most out of your prototypes.
- Extending the benefits of prototyping: Advice for those experienced in prototyping who have demanding customers, or who simply need a few new ideas.

In this e-toolkit

- Using paper prototypes to manage risk: Using a real-life scenario, the authors explain why and how paper prototypes can be ideal for risk management in a time crunch.
- Looking back on 16 years of paper prototyping: With all of the changes and advancements over the past 16 years, the authors explain why paper prototypes are relevant and beneficial in technology today.
- Looking at prototyping today: Forget paper, here is a look at how 3D printers are making prototyping fast and easy.
- Teamwork: It takes a village to give your customers what they want

Next article

In this e-toolkit

Storyboards

Storyboards help developers visualize the sequence and interconnectedness of their work. They allow for a "big picture" approach that may be very useful in requirements gathering.

- What is a storyboard?
- User Interface flow diagrams (storyboards): The Agile answer to storyboards.
- Use cases and stories: An explanation of the differences between use cases and the stories of Extreme Programming (XP).
- Increase business agility with some of these tips.
- Agile's growing up: Learn how Agile's maturity can impact an infrastructure and why it doesn't necessarily equate to success.

Next article

In this e-toolkit

Modeling

A model can be made according to Unified Modeling Language (UML) or according to domain-specific modeling. Or, models can consist of stick figures on a whiteboard. All of these methods have their advantages and disadvantages. Use these links to figure out what method is right for you.

- The value of modeling: This white paper is an introduction to modeling, why it should be employed and the very basics of beginning software modeling. (PDF)
- UML for the software developer, Part 1 -- Building classes: Concentrating on three UML model diagram types, the author demonstrate how UML classes relate to code. There are diagrams and code examples to aid the reader.
- UML for the software developer, Part 2 -- Mastering associations: The second part of this series explains how developers can understand the relationships among classes in UML.

In this e-toolkit

- Domain-specific modeling -- Welcome to the next generation of software modeling: The author lays out the shortcomings of UML and argues for DSM. By increasing abstraction, he argues, we reduce complexity.
- Test your knowledge of unified modeling language: Take this quiz to find out how equipped you are in the world of unified modeling languages.

Next article

In this e-toolkit

Modeling in the Agile methodology

View examples, overviews, and outlines of modeling from an Agile, standpoint.

- Agile requirements modeling example: A detailed outline of requirements modeling from an Agile standpoint.
- Software modeling on plain old whiteboards (POWs): The benefits of whiteboards may appeal to developers across the board. This article describes how to take good photos of whiteboards to preserve the information for when it is inevitably wiped off.
- Comparing the various approaches to modeling in software development: This is a clear, succinct overview of modeling approaches, paradigms and estimated adoption rates. A handy graph lays it all out.
- Simple tools for software modeling: Post-Its, index cards and whiteboards are extolled in this article, but CASE tools are discussed as well. The author lays out the benefits (and disadvantages) of CASE and offers suggestions for selecting CASE tools.

In this e-toolkit

State transition diagrams

State transition diagrams allow developers and users to see how a program might behave. This anticipation of events is useful when discussing requirements.

- What is a state diagram?
- State-transition diagrams: This article explains what state-transitions are and why they are important. There is also a series of questions for testing state-transition diagrams.
- Automating state-transitions: The Microsoft Developer's Network state-transitions within Visual Studio. Code examples aid the reader.
- Visual Requirements: There is a section devoted to state-transition diagrams in this article on diagrams in software development. The author provides a clear perspective on state diagrams and the necessary part they play among the other diagrams. (PDF)
- Understand testing diagrams: This article includes six questions that will help explain UML diagrams.

In this e-toolkit

Use Cases

State transition diagrams allow developers and users to see how a program might behave. This anticipation of events is useful when discussing requirements.

- What is a state diagram?
- State-transition diagrams: This article explains what state-transitions are and why they are important. There is also a series of questions for testing state-transition diagrams.
- Automating state-transitions: The Microsoft Developer's Network state-transitions within Visual Studio. Code examples aid the reader.
- Visual Requirements: There is a section devoted to state-transition diagrams in this article on diagrams in software development. The author provides a clear perspective on state diagrams and the necessary part they play among the other diagrams. (PDF)

Next article

In this e-toolkit

Tools to help with Agile requirements

Useful tools that can help you get the job done faster and more efficiently.

- Need drag and drop story mapping? Bauer has a JIRA plug-in tool for visual presentation and editing.
- Online sticky notes are another great option, particularly when teams are distributed. Lino allows everyone on the team to chime in.
- Other story mapping options include cardboardit and featuremap.

Next article

In this e-toolkit

Other useful resources on requirements gathering

Want more information? – Visit these links to learn more.

- Get the Agile scrum right the first time.
- Avoid the garbage in/garbage out problem in Agile.
- Here's where business requirements documents come in to play.
- It's all hands on deck when it comes to Agile requirements.
- *Writing Effective Use Cases*: Alistair Cockburn advises readers how to write -- as opposed to model -- use cases in this book.
- *Managing Software Requirements: A Unified Approach*: Dean Leffingwell and Don Widrig emphasize team skills in this book

Next article

In this e-toolkit

Getting more PRO+ exclusive content

This e-guide is made available to you, our member, through PRO+ Offers – a collection of free publications, training and special opportunities specifically gathered from our partners and across our network of sites.

PRO+ Offers is a free benefit only available to members of the TechTarget network of sites.

Take full advantage of your membership by visiting http://pro.techtarget.com/ProLP/

Images; Fotalia