DOWNLOAD PDF METHODS, TECHNIQUES AND TOOLS IN INFORMATION DESIGN PT. 2.

Chapter 1 : Project Management: Tools & Techniques

Methods, Techniques and Tools in Information Design Symposium on Human Interface , Held as Part of HCI International , Beijing, China, July , , Proceedings Part I Human Interface and the Management of Information.

There are tools like Rapise which can make your life easier, but how do you know which one to pick? Part 1 â€" Techniques Of course, efficient crossbrowser CSS development starts with techniques and good practices. Reset CSS Due to the fact web browsers define different default styling for html elements, the first thing to do is to always include a CSS reset in your stylesheet. Internet Explorer, especially the dinosaur IE6, is the front-end developer nightmare. In order to minimize their errors, Microsoft implemented conditionnal comments in their browser, which allow you to link a stylesheet that will be interpreted by a browser alone. The code below will only target Opera, allowing you to add CSS rules only for this browser. But it can happen sometimes. Here is a nice hack to write Safari-only CSS rules. Once you inclued JQuery and browserdetect. For example, your body tag will look like this if the visitor browser is Firefox 3: Simply paste it on the functions. Internet fucking Explorer 6 gives us, developers, lots of headaches. Although this is definitely not the best method to make your clients happy, it can be really fun to implement on your own website. You guessed it, the following line of code will make IE6 crash. Part 2 â€" Tools Techniques are the required knowledges for efficient cross browser development, but tools can make you save time, hasle, and in three words, make your life easier. For example, it apply new sizes for your boxes, according to the fact that IE have this well known box model bug. This is exactly Browsershots purpose: You select the OS, you select the browser, you wait a bit, and the site show you screenshots of your website under the selected OS and browsers. While Browsershots is obviously an usefull version, there are a few bad points. The worst of it is the waiting time for being able to see your screenshots, depending on how many browsers you selected. Run multiples version of IE on your PC MultipleIEs is a Windows-only program allowing you to install and run multiples versions of Internet Explorer from version 3 to version 6 on your computer. Sadly, it runs on X11, it is slow and buggy.

DOWNLOAD PDF METHODS, TECHNIQUES AND TOOLS IN INFORMATION DESIGN PT. 2.

Chapter 2 : 15+ techniques and tools for cross browser CSS coding

*Methods, Techniques and Tools in Information Design: Symposium on Human Interface , Held as Part I (Lecture Notes in Computer Science) [Michael J. Smith, Gavriel Salvendy] on calendrierdelascience.com *FREE* shipping on qualifying offers. This is the first of a two-volume set that constitutes the refereed proceedings of the Symposium on Human.*

After further investigation, the macro builder, as well as the PDF template analyzed through this research, have been observed in attacks associated to multiple groups. One commonality on the analysis is that all the groups observed until now are mainly targeting finance related institutions. Apart from Cobalt Gang, there are several financially motivated threat actors behind the infrastructure linked to this analysis. One of the most common approaches is the use of spear phishing emails employing social engineering or commonly used exploits such as CVE or the ThreadKit builder to trick the employees of organizations of interest. This approach makes it more difficult for threat hunters and defenders to find those needles in the haystack necessary to identify a campaign and its objectives. One of the groups well known for following these TTPs is the Cobalt Gang, which is still active even after the arrest of their alleged leader in Spain this year. As a result, we have been able to identify both the use of a common macro builder as well as specific document metadata which have allowed us to track and cluster new activity and infrastructure associated with these actors. A Recent Effective Example of Delivery One of the latest examples related to the campaign under analysis was used in attacks just a few days ago. It shows the simplicity of the attack delivery employed by these groups. The attack reinforces the fact that email is still one of the primary attack vectors we continuously observe. The sample shown in Figure 1 can already be found in popular public online malware repositories. Instead it seeks to use social engineering to convince the user click a link to download a malicious macro. PDF sample with embedded link The PDF is simple and embeds a link that will open a legitimate Google location, and redirect the browser to a malicious document from there: Malicious doc browser redirect In order to be effective against static analysis tools, the attackers crafted the PDF to seem more authentic: Keep in mind that PDFs with low number of pages or high entropy in the content can raise suspicious flags in static analysis. PDF static analysis Figure 5. PDF text used to fill pages By employing these two techniques the PDF avoids almost all traditional AV detection, resulting in a very effective transport of the first stage of the attack via email. If the attack progresses, the user will be taken to the download of an MS Word document containing malicious macros that has very low detection rate at the moment of this campaign delivery. From a metadata standpoint, the document does not include any specific signal or characteristic that would help us tracking documents from the same author, as shown in Figure 6. So, the question is nowâ€¦ how can this simple delivery method help identify the campaign and objectives? Macro Builder Identification The attack also achieves quite low detection results with its macro code, so one of the first focuses of the investigation is the identification of a possible underlying builder. The macro code is over lines in length, and starts declaring a set of variables with a very specific nomenclature in this sample, letXX num: Using specific variable format in decoding routines Procedures and functions are also defined using the same nomenclature in this sample, letXX: VBA pattern in other documents One initial approach to hunt for the pattern can be based on the following regular expressions for the different areas:

Chapter 3 : Design methods - Wikipedia

Methods, Techniques and Tools in Information Design: Symposium on Human Interface , Held as Part of HCI International , Beijing, China, July , , Proceedings, Part I / Edition 1 This is the first of a two-volume set that constitutes the refereed proceedings of the Symposium on Human Interface , held in Beijing, China in July.

ISD tools include at least a part of method knowledge. Typically tools contain parts of the conceptual structure as their schema definition, support modeling with certain notations, or support the process definition and management Odell Tool support is important for our research questions because tools can ensure that method knowledge is also applied and does not remain only as method descriptions i. Deep structure denotes those aspects of method knowledge which reflect the domain under development, whereas surface structure and physical structure deal with properties of modeling tools. Surface structure describes user-interface characteristics of an ISD tool, such as how method knowledge behind a modeling technique is visible in dialogs, menu commands and reports. This resembles the notational part of method knowledge. Physical structure denotes the technical means applied in the implementation of the ISD tool. In this section our focus is on tools which support the use of methods, i. This formed the third italicized part in our definition of ISD cf. Second, we describe relationships between methods and tools in more detail through the concept of method-tool companionship. This allows us to explain how tools can support modeling techniques. This is relevant for our research questions, since we seek to apply metamodels in specifying modeling techniques enacted by ISD tools. Thus, it is possible to describe the underlying elements of methods i. This focus also means that we believe that the use of metamodeling in local method development is most beneficial when related to customization of tools. Naturally, metamodeling can be applied for reasons other than local method development cf. Brinkkemper , but local method development aiming only to specify and compare methods takes us only half-way, because the usefulness of a method is realized only when it is applied. Using metamodels without considering their support in ISD tools would be the same as designing an IS without implementing it. Technological developments have lead to a large number of tools that cover nearly all tasks of ISD. At the same time the term CASE Computer-Aided System Engineering has been extended to denote all types of computer tools from business modeling and requirements capture to implementation tools. The concept of CASE is broad and it includes compilers, project management tools, and even editors [10]. In this thesis we examine CASE tools and methods in the context of modeling. These modeling tools are usually used to support early phases of ISD. As already mentioned, the term method is restricted in this thesis to mean that part of the method knowledge that it is possible to capture into a formalized part of a tool. Types of methods and tools deployed during different phases of ISD are described in Table As shown in the table, support for business process re-engineering and development include both methods and tools cf. On the method side, process maps, workflow models, task structures and action diagrams are applied Harrington , Goldkuhl, , Lundeberg, On the tool side, computing power is applied for example to benchmark, compare, and simulate business processes through models. The methodical aspects of these tools rely on brain-storming, interviews and cooperation. In the system analysis and design phases the upper-CASE tools support methods such as conceptual data modeling ER models and derivatives and structured analysis and design e. Most CASE products nowadays focus on supporting object-oriented methods, and recently tool support has been extended towards business modeling Wangler et al. In this thesis we also concentrate on business modeling methods which, to a large extent, lack computer support Stegwee and Van Waes The relationship between methods and tools is most obvious in the construction phase: The availability of compilers renders programming methods and languages practicable, because there is little point in writing first in some programming language and then making a translation by hand. During construction and maintenance, computer aided tools can support version control, configuration management, and reverse engineering. On the one hand, tools mechanize operations prescribed by methods by storing system representations, transforming representations

from one type of model to another, and displaying representations in varying forms. On the other hand, tools empower users by enhancing correctness checking and analytical power, by freeing them from tedious documentation tasks, and by providing multi-user coordination access and version control. None of these features could be easily available in manual method use. The companionship between tools and methods has also evolved in response to technical innovations Norman and Chen These require extensions to existing methods or entirely new types of methods to support their development e. CASE tools do not provide the same level of support for all types of method knowledge. For example, there are more tools that support model building, representation and checking than there are tools that guide processes or provide group support Tolvanen et al. Naturally, some aspects of methods lend themselves more easily to automation than others Olle et al. Nevertheless some method knowledge need to be present in an ISD tool. The presence of methods can also be viewed using CASE tool support functionality, i. In the following these are discussed based on support for four different design steps Olle et al. The majority of steps in design deal with abstractions, and thus it is also the most supported step Olle et al. On the level of method-tool companionship this requires that a tool includes all the modeling related parts of the conceptual structure and employs notational representations for them in modeling editors. Hence, this design step can be carried out only after some aspects of the object system have been abstracted into models. Checking operates mostly on the conceptual structure and deals with constraints and rules of the method also called verification rules Wijers Although some checking activities can be carried out by using alternative representation forms, such as matrixes for cross-checking, checking operates mostly on the non-notational concepts. Therefore, to achieve companionship this requires that the conceptual structure of the method includes not only concepts related directly to representation i. For example, in most object-oriented methods, the link between a state model and a class in a class model is vaguely defined one good exception is Embley et al. A state model can include states from several classes and therefore a tool can not analyze whether all attributes of the class have values during its life-cycle. To carry out this type of checking, the method specifications should include a reference from each state to a corresponding class, or have state models that are used for instances i. As a result, we do not have many methods which are developed specially for CASE environments and take full advantage of automation. Furthermore, in methods which apply multiple modeling techniques, the need for checking is stressed. Also, if multiple tools are used, method integration is a prerequisite of successful tool integration. During a form conversion an underlying conceptual structure, a notation, or a representation form changes. Examples of such conversions, found in many CASE tools, are model analysis, reporting functions, and code generation. To support these, the conceptual structure should include types and constraints which are not necessarily required for the abstraction or checking steps. For example, to generate program code e. These constructs are often missing from conceptual structures of text-book methods. As a result, CASE vendors need to extend methods in order to provide additional tool functionality. It should be noted that not all conversions can be fully automated, but rather often require human interaction. Because the review step is often carried out together with the users or experts in the object system domain, the notation part of method knowledge is emphasized here. To ensure that models describe all relevant parts of the system, the notation should be sufficient to represent them. Naturally, the adequate support of the notation reflects the underlying conceptual structure. Since the effectiveness of the tool is always dependent on the method it is important how a method or its parts are implemented in a tool. In other words, which aspects and which level of detail of method knowledge are supported. In our research, this means that the applicability of methods is partly dictated by how well the tool supports their techniques. Hence, method-tool companionship is based mainly on supporting the conceptual structure and its related notation, and secondly the modeling process and design objectives. The modeling process is relevant because the user interface i. The design objectives are relevant to method-tool companionship because tools should also support generation of design alternatives or produce solutions automatically. Tool developers have concentrated more on producing technical solutions such as repositories and intelligent knowledge-based support in their products, while the methodical part has been given a lower

priority. Furthermore, methods which have been coded as a part of a tool, what we call method-dependent CASE, do not allow the further development or extension of methods according to the situation specific needs. Against this backdrop the surprisingly slow diffusion of CASE tools is also more understandable. Research into introducing CASE in an organization reveals that the main problems in the introduction are not the technical changes, but the methodical and cultural changes which the use of the new tool will inevitably cause Aaen et al. These observations are obvious, because the effective use of CASE tools is not possible without an adequate experience and knowledge of method use Humphrey Introducing method-dependent CASE tools causes changes in the way of working and in the use of methods. In contrast to the tool-driven approach, one should select tools so that they fit into the local domain and ISD situations. Several studies of CASE tools see e. Whereas these researchers have pointed out the demand for flexible CASE support, the technological point of view has still been dominant. This problem is discussed from the viewpoint of tool adaptation in Chapter 3. It is also possible to classify tools based on the level of integration:

DOWNLOAD PDF METHODS, TECHNIQUES AND TOOLS IN INFORMATION DESIGN PT. 2.

Chapter 4 : Information mapping - Wikipedia

In addition to the articles on this current page, see the following blogs which have posts related to Methods of Data Collection. Scan down the blog's page to see various posts. Also see the section "Recent Blog Posts" in the sidebar of the blog or click on "next" near the bottom of a post in the blog.

Background[edit] Design traditionally has been associated with expression and production. Social, political and economic developments of the late 19th and first half of the 20th century put into motion modern benefits and constraints for living and working. Industrial and technological breakthroughs associated with this period created social and economic complexities for people and their environment. Disciplines such as architecture, urban planning, engineering and product development began to tackle new types of problem-solving past traditional artifact making. More informed and methodical approaches to designing were required. They believed that a single craft-based designer producing design solutions was not compatible with addressing the evolving complexity of post-industrial societies. The [6] key benefit was to find a method that suits a particular design situation. Where process meets method[edit] When process and method are discussed, they tend to be used interchangeably. However, while they are two sides to the same coin, they are different. Process contains a series of actions, events, mechanisms, or steps, which contain methods. Method is a way of doing something, especially a systematic way through an orderly arrangement of specific techniques. Each method has a process. Design Methods is challenging to implement since there are not enough agreed-upon tools, techniques and language for consistent knowledge transfer. While there are many conceptual models and frameworks, there needs to be more granularity of tools and techniques. There are also many variables that affect outcomes since logic and intuition interplay with one another. Two people can therefore use the same method and arrive at different outcomes. Three "camps" seemed to emerge to integrate the initial work in Design Methods: Behaviorism interpreted Design Methods as a way to describe human behavior in relation to the built environment. Its clinical approach tended to rely on human behavior processes taxonomic activities. Reductivism broke Design Methods down into small constituent parts. This scientific approach tended to rely on rationalism and objectified processes such as epistemological activities. Phenomenology approached design methods from an experiential approach human experience and perception. The purpose of the Society is to promote "the study of and research into the process of designing in all its many fields" and is an interdisciplinary group with many professions represented, but all bound by the conviction of the benefits of design research. Jones and Christopher also questioned their original thesis about design methods. An interesting shift that affected design methods and design studies was the lecture from Herbert A. Simon , the Nobel laureate, who presented "The Sciences of the Artificial. Nigel Cross has been prolific in articulating the issues of design methods and design research. The discussion of the ongoing debate of what is design research and design science was, and continues to be articulated by Cross. His thesis is that design is not a science, but is an area that is searching for "intellectual independence. Scientific method was borrowed as one framework, and the term "design science" was coined in at the Second Conference on the Design Method focusing on a systematic approach to practicing design. Cross defined the "science of design" as a way to create a body of work to improve the understanding of design methodsâ€”and more importantly that design methods does not need to be a binary choice between science and art. Nigan Bayazit, professor at the Istanbul Technical University, published an overview of the history of design methods. She stated that "Design methods people were looking at rational methods of incorporating scientific techniques and knowledge into the design process to make rational decisions to adapt to the prevailing values, something that was not always easy to achieve. The physical embodiment of man-made things, how these things perform their jobs, and how their users perceive and employ them Construction as a human activity, how designers work, how they think, and how they carry out design activity, and how non-designers participate in the process What is achieved at the end of a purposeful design activity, how an artificial thing appears, and what it means Embodiment of configurations

Systematic search and acquisition of knowledge related to design and design activity Significance of emergence of design research and design studies[edit] Both research and design studies made design more visible and accountable. Research was recognized at the outset by design methods as a type of leg-work to The eventual debate about design methods and whether design is an art or science is not a new. Partisans on both sides of the issue have framed it as a binary choice of something to lose or gain. However, this false argument was viewed by John Chris Jones, who recognized the "logical, systematic, behavioristic, operational aspects of new methods" which could be viewed as science might be seen as "anti-life" which treat people as "instruments. Jones sought to bring both together and act as checks-and-balances for design methods. Jones viewed methodology as "mere symbolic contrivances" and "would lose its value" if it did not reflect "the personal issues which matter most to the people who will take decisions. America was also a magnet for practicing design professionals to codify their successes in design practice and backing into larger theories about the dynamics of design methods. American designers were much more pragmatic at articulating design methods and creating an underlying language about the practice of industrial and graphic design. They were tied to economic systems that supported design practice and therefore focused on the way design could be managed as an extension of business, rather than the European approach to design methods based on transforming engineering by design. Industrial design was the first area that made inroads into systematizing knowledge through practice. Raymond Loewy was instrumental at elevating the visibility of industrial design through cult of personality appearing three times on front cover of Time Magazine. Henry Dreyfuss had a profound impact on the practice of industrial design by developing a systematic process used to shape environments, transportation , products and packaging. His focus on the needs of the average consumer was most celebrated in his book Designing for People, an extensive exploration of ergonomics. In seven pages, Doblin presents a straightforward and persuasive argument for design as a systematic process. He described the emerging landscape of systematic design: For large complex projects, it "would be irresponsible to attempt them without analytical methods" and rallied against an "adolescent reliance on overly intuitive practices. Doblin and others were responding to the increased specialization of design and the complexity of managing large design programs for corporations. It was a natural process to begin to discuss how design should move upstream to be involved with the specifications of problems, not only in the traditional mode of production which design had been practiced. Particularly since , design methods and its intersection with business development have been visibly championed by numerous consultancies within design industry. The continuity of approaches to design projects by such representative firms is the generation of inputs incited by the human condition in varied contexts. These approaches utilize a sustainable methods-based mode of making that takes into account critical analytic and synthetic skills toward more informed and inspired specifications grounded in: Direct investigation of human circumstances to draw out impressions Engagement by client-side and end-user participants in design process Open articulation by practitioners of multiple disciplines facilitated by design Significance of role of professional design practice[edit] Practitioners approached design methods from a different angle than John Christopher Jones and the group of engineers and designers who convened in Many practitioners, through actual design opportunities, began to confront the complexities of the market and clients. They began to address issues of specifications, users , distribution and innovation. Since there were no established methods, each practitioner began to develop frameworks and languages to describe a new way to design. Like any market-based model, there were many competing ideas about these new methods and their basis. Many of these designers may have been aware of the design methods movement, but many were not. Yet all their ideas were aligned to many of the basic tenets of the conference which advocated a more rigorous way of doing design. However, the social perspectives and criticisms of mediocre products of participants may not have been shared or agreed with. While this relationship has been identified, it has not been universally recognized or accepted by diverse design communities. Designers have strong connection not only to clients but also to end users who consume products and services. Design as a function within corporations , or as independent consultancies, have not always collaborated well with business. Clients and the market have

traditionally viewed design as an expressive and production function, rather than as a strategic asset. Designers have focused their skills and knowledge in creating designed artifacts, and indirectly addressed larger issues within this creative process. They have been uneasy about articulating their value to business in terms that business executives could understand. There were moves to bridge this gap. In England , the British Design Council now called the Design Council was founded in by the British wartime government as the Council of Industrial Design with the objective "to promote by all practicable means the improvement of design in the products of British industry". The Design Management Institute is an international nonprofit organization that seeks to heighten awareness of design as an essential part of business strategy. Founded in , DMI has become the leading resource and international authority on design management. Alternative view[edit] Some designers and design historians have challenged, even rejected, the idea that design supports the goals and objectives of the economic systems they find themselves in. Victor Papanek â€" was a trail blazer in the definition of sustainable design and addressing social issues through design. His book Design for the Real World in the late s articulated a world for design to use less resources and address local social issues for ecologically sound design to serve the poor, the disabled and the elderly. The disciplines of sustainable design and universal design are echoed here. Professor of design history at the University of Illinois at Chicago Victor Margolin addressed the inherent role of design communities supporting an economic system, which he called the "expansion model", where "the world consists of markets in which products function first and foremost as tokens of economic exchange. They attract capital which is either recycled back into more production or becomes part of the accumulation of private or corporate wealth. If the elements of this system are damaged or thrown out of balance or if essential resources are depleted, the system will suffer severe damage and will possibly collapse. John Chris Jones recognized the role of business, as one stakeholder among many, but did not view design methods as a business management tool. Design management focuses on how to define design as a business function and provides a language and method of how to effectively manage it. Proliferation of information technologies[edit] Internet businesses realized early that technologists alone were not going to create "killer apps" that would win customers. Together they had to rapidly accelerate time-to-value and learn how to do things that had little precedent. Other professionals were incorporated from cognitive science , ethnography , and library science to name a few. User-centric approaches were developed resulting in the creation of whole workflow systems to accommodate diversity in skills and tools. These diverse groups brought markedly different languages and models native to their disciplines which posed significant integration-challenges, including hours, in determining how to work together. Clement Mok , founder of Studio Archetype acquired by Sapient , recognized this trend and began to articulate the new professional design situation being agitated by new information technologies marked by the Internet and advancements in computing media. He described a multi-media landscape that was converging into an integrated digital space. Adjacent to this was the redefinition of skills and roles that would create, build, sustain, and innovate this dynamic environment. In his book, Designing Business, [11] Mok emphasized redefinition of design practice dramatically affected by technological change: They were 30 years ahead of the expansion of the Internet and explained the basic premise of its value by stating: The nett sic effect is expected to be one of mutual stimulation in which open minded people and programmes nudge each other into unpredictable, novel but realistic explorations John Chris Jones recognized this by stating: The language of the conversation must bridge the logical gap between past and future, but in doing so it should not limit the variety of possible futures that are discussed nor should it force the choice of a future that is unfree. The key benefit is to find a method that suits a particular design situation. Emphasis on the user Use of basic research methods to validate convictions with fact Use of brainstorming and other related means to break mental patterns and precedent Increased collaborative nature of design with other disciplines A large challenge for design as a discipline, its use of methods and an endeavor to create shared values, is its inherent synthetic nature as an area of study and action. This allows design to be extremely malleable in nature, borrowing ideas and concepts from a wide variety of professions to suit the ends of individual practitioners. It also makes design vulnerable

since these very activities make design a discipline unextensible as a shared body of knowledge. Schon foresaw the increasing instability of traditional knowledge and how to achieve it.

DOWNLOAD PDF METHODS, TECHNIQUES AND TOOLS IN INFORMATION DESIGN PT. 2.

Chapter 5 : Information system development tools

Human Interface and the Management of Information. Methods, Techniques and Tools in Information Design Symposium on Human Interface , Held as Part of HCI International , Beijing, China, July , , Proceedings, Part I.

Solving Engineering Problems in Dynamics An organization needs to define some standard of problem solving, so that leadership can effectively direct others in the research and resolution of issues. In problem solving, there are four basic steps. Define the problem Diagnose the situation so that your focus is on the problem, not just its symptoms. Helpful techniques at this stage include using flowcharts to identify the expected steps of a process and cause-and-effect diagrams to define and analyze root causes. The chart below identifies key steps for defining problems. These steps support the involvement of interested parties, the use of factual information, comparison of expectations to reality and a focus on root causes of a problem. Review and document how processes currently work who does what, with what information, using what tools, communicating with what organizations and individuals, in what time frame, using what format, etc. Generate alternative solutions Postpone the selection of one solution until several alternatives have been proposed. Having a standard with which to compare the characteristics of the final solution is not the same as defining the desired result. A standard allows us to evaluate the different intended results offered by alternatives. Considering multiple alternatives can significantly enhance the value of your final solution. Brainstorming and team problem-solving techniques are both useful tools in this stage of problem solving. Many alternative solutions should be generated before evaluating any of them. If we focus on trying to get the results we want, we miss the potential for learning something new that will allow for real improvement. Evaluate and select an alternative Skilled problem solvers use a series of considerations when selecting the best alternative. They consider the extent to which: A particular alternative will solve the problem without causing other unanticipated problems. All the individuals involved will accept the alternative. Implementation of the alternative is likely. The alternative fits within the organizational constraints. The most effective approach, by far, has been to involve others in the implementation as a way of minimizing resistance to subsequent changes. Feedback channels must be built into the implementation of the solution, to produce continuous monitoring and testing of actual events against expectations. Problem solving, and the techniques used to derive elucidation, can only be effective in an organization if the solution remains in place and is updated to respond to future changes. Dennis Beecroft, Grace L. Duffy, and John W.

DOWNLOAD PDF METHODS, TECHNIQUES AND TOOLS IN INFORMATION DESIGN PT. 2.

Chapter 6 : Problem Solving Steps & Process - Learning Resources | ASQ

The papers of this first volume address the following major topics: design and evaluation methods and techniques, visualizing information, retrieval, searching, browsing and navigation, development methods and techniques, as well as advanced interaction technologies and techniques.

Bourgeois Learning Objectives Upon successful completion of this chapter, you will be able to: Introduction When someone has an idea for a new function to be performed by a computer, how does that idea become reality? If a company wants to implement a new business process and needs new hardware or software to support it, how do they go about making it happen? In this chapter, we will discuss the different methods of taking those ideas and bringing them to reality, a process known as information systems development. Programming As we learned in chapter 2, software is created via programming. Programming is the process of creating a set of logical instructions for a digital device to follow using a programming language. True, sometimes a programmer can quickly write a short program to solve a need. But most of the time, the creation of software is a resource-intensive process that involves several different groups of people in an organization. In the following sections, we are going to review several different methodologies for software development. This methodology was first developed in the s to manage the large software projects associated with corporate systems running on mainframes. It is a very structured and risk-averse methodology designed to manage large projects that included multiple programmers and systems that would have a large impact on the organization. In this phase, a review is done of the request. Is creating a solution possible? What is currently being done about it? Is this project a good fit for our organization? A key part of this step is a feasibility analysis, which includes an analysis of the technical feasibility is it possible to create this? This step is important in determining if the project should even get started. In this phase, one or more system analysts work with different stakeholder groups to determine the specific requirements for the new system. No programming is done in this step. Instead, procedures are documented, key players are interviewed, and data requirements are developed in order to get an overall picture of exactly what the system is supposed to do. The result of this phase is a system-requirements document. It is in this phase that the business requirements are translated into specific technical requirements. The design for the user interface, database, data inputs and outputs, and reporting are developed here. The result of this phase is a system-design document. This document will have everything a programmer will need to actually create the system. The code finally gets written in the programming phase. The result of this phase is an initial working program that meets the requirements laid out in the system-analysis phase and the design developed in the system-design phase. In the testing phase, the software program developed in the previous phase is put through a series of structured tests. The first is a unit test, which tests individual parts of the code for errors or bugs. Next is a system test, where the different components of the system are tested to ensure that they work together properly. Finally, the user-acceptance test allows those that will be using the software to test the system to ensure that it meets their standards. Any bugs, errors, or problems found during testing are addressed and then tested again. Once the new system is developed and tested, it has to be implemented in the organization. This phase includes training the users, providing documentation, and conversion from any previous system to the new system. Implementation can take many forms, depending on the type of system, the number and type of users, and how urgent it is that the system become operational. These different forms of implementation are covered later in the chapter. This final phase takes place once the implementation phase is complete. In this phase, the system has a structured support process in place: The SDLC methodology is sometimes referred to as the waterfall methodology to represent how each step is a separate part of the process; only when one step is completed can another step begin. After each step, an organization must decide whether to move to the next step or not. This methodology has been criticized for being quite rigid. For example, changes to the requirements are not allowed once the process has begun. No software is available until after the programming phase. Again, SDLC was developed

for large, structured projects. Projects using SDLC can sometimes take months or years to complete. Because of its inflexibility and the availability of new programming techniques and tools, many other software-development methodologies have been developed. Many of these retain some of the underlying concepts of SDLC but are not as rigid. Rapid Application Development The RAD methodology Public Domain Rapid application development RAD is a software-development or systems-development methodology that focuses on quickly building a working model of the software, getting feedback from users, and then using that feedback to update the working model. After several iterations of development, a final version is developed and implemented. The RAD methodology consists of four phases: This phase is similar to the preliminary-analysis, system-analysis, and design phases of the SDLC. In this phase, the overall requirements for the system are defined, a team is identified, and feasibility is determined. In this phase, representatives of the users work with the system analysts, designers, and programmers to interactively create the design of the system. One technique for working with all of these various stakeholders is the so-called JAD session. JAD is an acronym for joint application development. A JAD session gets all of the stakeholders together to have a structured discussion about the design of the system. Application developers also sit in on this meeting and observe, trying to understand the essence of the requirements. In the construction phase, the application developers, working with the users, build the next version of the system. This is an interactive process, and changes can be made as developers are working on the program. This step is executed in parallel with the User Design step in an iterative fashion, until an acceptable version of the product is developed. In this step, which is similar to the implementation step of the SDLC, the system goes live. All steps required to move from the previous state to the use of the new system are completed here. Many of the SDLC steps are combined and the focus is on user participation and iteration. This methodology is much better suited for smaller projects than SDLC and has the added advantage of giving users the ability to provide feedback throughout the process. SDLC requires more documentation and attention to detail and is well suited to large, resource-intensive projects. RAD makes more sense for smaller projects that are less resource-intensive and need to be developed quickly. Agile Methodologies Agile methodologies are a group of methodologies that utilize incremental changes with a focus on quality and attention to detail. Each increment is released in a specified period of time called a time box , creating a regular release schedule with very specific objectives. While considered a separate methodology from RAD, they share some of the same principles: The characteristics of agile methods include: The goal of the agile methodologies is to provide the flexibility of an iterative approach while ensuring a quality product. Lean Methodology The lean methodology click to enlarge One last methodology we will discuss is a relatively new concept taken from the business bestseller The Lean Startup , by Eric Reis. In this methodology, the focus is on taking an initial idea and developing a minimum viable product MVP. The MVP is a working software application with just enough functionality to demonstrate the idea behind the project. Once the MVP is developed, it is given to potential users for review. Feedback on the MVP is generated in two forms: Using these two forms of feedback, the team determines whether they should continue in the same direction or rethink the core idea behind the project, change the functions, and create a new MVP. This change in strategy is called a pivot. Several iterations of the MVP are developed, with new functions added each time based on the feedback, until a final product is completed. The biggest difference between the lean methodology and the other methodologies is that the full set of requirements for the system are not known when the project is launched. As each iteration of the project is released, the statistics and feedback gathered are used to determine the requirements. The lean methodology works best in an entrepreneurial environment where a company is interested in determining if their idea for a software application is worth developing. The Quality Triangle The quality triangle When developing software, or any sort of product or service, there exists a tension between the developers and the different stakeholder groups, such as management, users, and investors. This tension relates to how quickly the software can be developed time , how much money will be spent cost , and how well it will be built quality. The quality triangle is a simple concept. So what does it mean that you can only address two of the three?

DOWNLOAD PDF METHODS, TECHNIQUES AND TOOLS IN INFORMATION DESIGN PT. 2.

However, if you are willing or able to spend a lot of money, then a project can be completed quickly with high-quality results through hiring more good programmers. Of course, these are just generalizations, and different projects may not fit this model perfectly. But overall, this model helps us understand the tradeoffs that we must make when we are developing new products and services. Programming Languages As I noted earlier, software developers create software using one of several programming languages. A programming language is an artificial language that provides a way for a programmer to create structured code to communicate logic in a format that can be executed by the computer hardware. Over the past few decades, many different types of programming languages have evolved to meet many different needs. In these early languages, very specific instructions had to be entered line by line â€" a tedious process. First-generation languages are called machine code. In machine code, programming is done by directly setting actual ones and zeroes the bits in the program using binary code. Assembly language gives english-like phrases to the machine-code instructions, making it easier to program. An assembly-language program must be run through an assembler, which converts it into machine code. Here is an example program that adds and using assembly language: Most third-generation languages must be compiled, a process that converts them into machine code.

DOWNLOAD PDF METHODS, TECHNIQUES AND TOOLS IN INFORMATION DESIGN PT. 2.

Chapter 7 : Overview of Basic Methods to Collect Information

Information system development tools The shell model allows us to illustrate the tool support addressed in this thesis: ISD tools include at least a part of method knowledge. Typically tools contain parts of the conceptual structure as their schema definition, support modeling with certain notations, or support the process definition and.

Estimate project costs and schedules. Establish a dependable project control and monitoring system. Tools Project management is a challenging task with many complex responsibilities. Fortunately, there are many tools available to assist with accomplishing the tasks and executing the responsibilities. Some require a computer with supporting software, while others can be used manually. Project managers should choose a project management tool that best suits their management style. No one tool addresses all project management needs. Both of these project management tools can be produced manually or with commercially available project management software. Both charts display the total project with all scheduled tasks shown in sequence. The displayed tasks show which ones are in parallel, those tasks that can be performed at the same time. The activities are the tasks of the project. The milestones are the events that mark the beginning and the end of one or more activities. Determine the proper sequence of activities. This step may be combined with 1 above since the activity sequence is evident for some tasks. Other tasks may require some analysis to determine the exact order in which they should be performed. Construct a network diagram. Using the activity sequence information, a network diagram can be drawn showing the sequence of the successive and parallel activities. Arrowed lines represent the activities and circles or "bubbles" represent milestones. Estimate the time required for each activity. Weeks are a commonly used unit of time for activity completion, but any consistent unit of time can be used. For each activity, the model usually includes three time estimates: Optimistic time - the shortest time in which the activity can be completed. Most likely time - the completion time having the highest probability. Pessimistic time - the longest time that an activity may take. From this, the expected time for each activity can be calculated using the following weighted average: Determine the critical path. The critical path is determined by adding the times for the activities in each sequence and determining the longest path in the project. The critical path determines the total calendar time required for the project. The amount of time that a non-critical path activity can be delayed without delaying the project is referred to as slack time. If the critical path is not immediately obvious, it may be helpful to determine the following four times for each activity: The earliest start and finish times of each activity are determined by working forward through the network and determining the earliest time at which an activity can start and finish considering its predecessor activities. The latest start and finish times are the latest times that an activity can start and finish without delaying the project. LS and LF are found by working backward through the network. The critical path then is the path through the network in which none of the activities have slack. The variance in the project completion time can be calculated by summing the variances in the completion times of the activities in the critical path. Given this variance, one can calculate the probability that the project will be completed by a certain date assuming a normal probability distribution for the critical path. The normal distribution assumption holds if the number of activities in the path is large enough for the central limit theorem to be applied. Update the PERT chart as the project progresses. As the project unfolds, the estimated times can be replaced with actual times. In cases where there are delays, additional resources may be needed to stay on schedule and the PERT chart may be modified to reflect the new situation. An example of a PERT chart is provided below: Improved forecasting of resource requirements. Identification of repetitive planning patterns which can be followed in other projects, thus simplifying the planning process. Ability to see and thus reschedule activities to reflect interproject dependencies and resource limitations following know priority rules. It also provides the following: Gantt charts are used to show calendar time task assignments in days, weeks or months. The tool uses graphic representations to show start, elapsed, and completion times of each task within a project. Gantt charts are ideal for tracking progress. The number of days actually required to

complete a task that reaches a milestone can be compared with the planned or estimated number. The actual workdays, from actual start to actual finish, are plotted below the scheduled days. This information helps target potential timeline slippage or failure points. These charts serve as a valuable budgeting tool and can show dollars allocated versus dollars spent. For each task, show the earliest start date, estimated length of time it will take, and whether it is parallel or sequential. If tasks are sequential, show which stages they depend on. Head up graph paper with the days or weeks through completion. Plot tasks onto graph paper. Show each task starting on the earliest possible date. Draw it as a bar, with the length of the bar being the length of the task. Above the task bars, mark the time taken to complete them. Schedule them in such a way that sequential actions are carried out in the required sequence. Ensure that dependent activities do not start until the activities they depend on have been completed. Where possible, schedule parallel tasks so that they do not interfere with sequential actions on the critical path. While scheduling, ensure that you make best use of the resources you have available, and do not over-commit resources. Also, allow some slack time in the schedule for holdups, overruns, failures, etc. In the final version of your Gantt chart, combine your draft analysis 3 above with your scheduling and analysis of resources 4 above. This chart will show when you anticipate that jobs should start and finish. An example of a Gantt chart is provided below: Benefits of using a Gantt chart include: Makes it easy to develop "what if" scenarios. Enables better project control by promoting clearer communication. Becomes a tool for negotiations. Shows the actual progress against the planned schedule. Can report results at appropriate levels. Allows comparison of multiple projects to determine risk or resource allocation. Rewards the project manager with more visibility and control over the project. The Future Project management tools have evolved from simple spreadsheet products to sophisticated, Web-based project information portals. The obvious trend in project management software, as with almost everything in information technology, is a move toward Web-based systems. Most project management tools can be accessed via browsers and those that do not currently have this capability are moving in that direction. The product acts as a portal development tool that allows firms to view information from products such as Microsoft Project over the Web. These project management portals are becoming more common as the collaboration capabilities of project management tools improve. WorkLenz is a software application that serves as a virtual project manager with intelligent agent features. The general direction is toward more integrated process and knowledge management systems, and user interfaces with a "Web" look. It is important for project management software vendors to keep things simple and easy to use.

Chapter 8 : Chapter Information Systems Development â€“ Information Systems for Business and Beyond

A wireframe is a two-dimensional illustration of a page's interface that specifically focuses on space allocation and prioritization of content, functionalities available, and intended behaviors.

Chapter 9 : Read e-book online Collaboration in Creative Design: Methods and Tools PDF - calendrierdela

Multi-finger Haptic interplay provides a landscape of applied sciences and techniques for multi-finger haptic interplay, including an research of the advantages and implications of including multiple-fingers to haptic functions. study subject matters lined include:design and keep watch over of complex haptic devices;multi-contact aspect.