

Ingeniería de Requisitos: Análisis y Gestión

Edgar Sarmiento Calisaya

Escuela Profesional de Ciencia de la Computación
Universidad Nacional de San Agustín de Arequipa,
Arequipa, Perú

Contenido

1. Ingeniería de Requisitos

- Elicitación
 - Léxico Extendido del Lenguaje
- Especificación
 - Usando Lenguaje Natural
 - Usando Modelos
- Análisis
 - Verificación
 - Validación
- Gestión

2. Herramientas

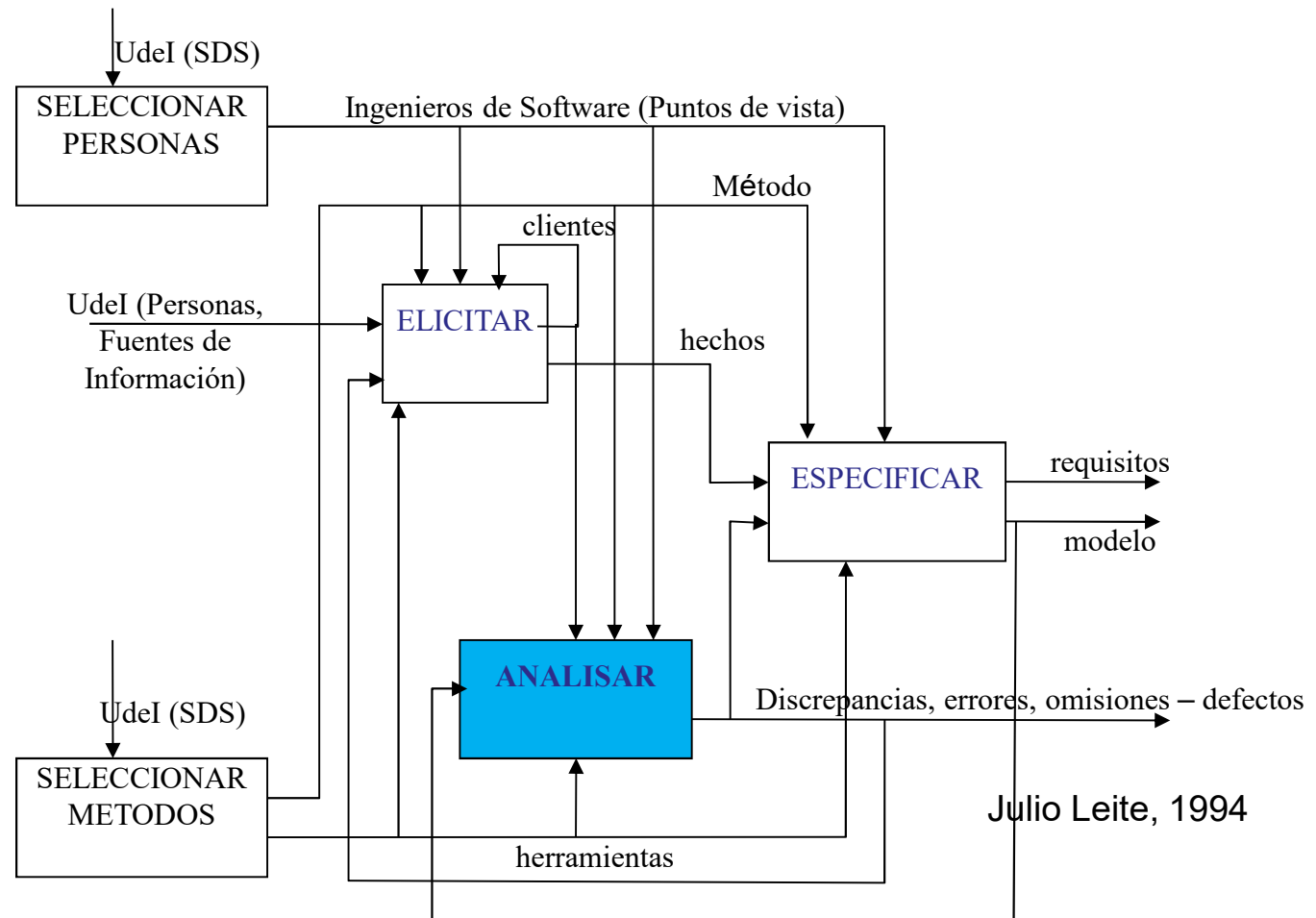
3. Gestión de Proyectos por Requisitos

4. Conclusión

Actividades de la Ingeniería de Requisitos

1

Principales Actividades de la Ingeniería de Requisitos



1.3 Análisis

- Fundamental para el éxito del proceso de desarrollo del software.
- Implica **tres actividades principales**:
 - **Identificación de partes**
 - **Verificación**
 - **Validación**

1.3 Análisis

REALIZA Identificación de Partes

REALIZA Verificación

REALIZA Validación

USA Personal

USA Métodos

USA Herramientas

DEPENDE DE Puntos de Vista

Identificación de Partes::

Organización

Almacenamiento

1.3 Análisis

- **Verificación:** **software** bien construido, libre de defectos?
 - Análisis usando métodos formales, como Z y VDM
 - **Inspección**
 - Checklists
 - Simulación
 - ¿Estamos construyendo el software correctamente?
- **Validación:** software satisface las necesidades reales del usuario?
 - Uso de comprobación informal
 - Uso de **prototipos** (p.e. ~~Design Sprint~~)
 - Reutilización de dominios
 - Uso de puntos de vista
 - **CON LA PARTICIPACIÓN DEL USUARIO!**
 - ¿Estamos construyendo el producto correcto?

Objetivo:
Especificación
libre de
Discrepancias
(conflictos),
Errores,
Omisiones –
DEFECTOS

Inspección

- El **objetivo** de la inspección es verificar si el producto producido por un proceso de producción está de acuerdo con los criterios de éxito del proceso.
- LAS INSPECCIONES SON FUNDAMENTALES PARA EL ASEGURAMIENTO DE LA CALIDAD
- La inspección **Fagan**, cuyo nombre proviene de su inventor Michael E. Fagan, ingeniero de **IBM** en los años setenta, es sin duda la técnica más potente para la **revisión** de documentos, contratos, código, etc.
- **Objetivo** de las inspecciones Fagan:
 - Encontrar tempranamente los defectos.
 - Prevenir el mal funcionamiento de los procesos o planes establecidos.
 - Proporcionar mejoras en la fiabilidad, disponibilidad y factibilidad del mantenimiento de software.
 - Descubrir continuamente la información técnica, asociada con las funciones, formularios, actividades internas que aseguran el producto.
 - Continuar el mejoramiento del proceso de desarrollo.
 - Establecer una igualdad de conocimiento dentro de los desarrolladores para la buena práctica de estándares y técnicas de desarrollo.

Inspección

- Un **equipo** de inspección de **Fagan** consiste en: un **asesor**, un **lector**, un **inspector**, y un **autor**.
 - El **asesor** debe asegurarse de que la preparación del equipo se centre en la detección de defectos sin desviarse.
 - El papel del **lector** es explicar y leer el producto que será inspeccionado a un paso razonable, pues el procedimiento se puede hacer de manera rápida, pero tal vez no se inspeccione lo que se desea.
 - El papel de un **inspector** es **examinar** el software desde el punto de vista de un usuario.
 - La presencia del **autor** en las reuniones de la inspección, generalmente se considera beneficiosa la presencia del autor porque puede asistir al equipo de la inspección para entender mejor el producto, además está preparado para entender la naturaleza de los defectos en los hallazgos del equipo.

Inspección

☉ El **método Fagan** establece 6 **pasos**:

Planificación	Cuando los materiales para ser inspeccionados pasan por los criterios de entrada (por ejemplo, el código fuente compila con éxito sin errores de sintaxis), miembros del equipo de inspección se seleccionan, y se establecen los horario de la inspección (por ejemplo, tiempo y lugar).
Visión General	Se dan instrucciones previas a los miembros del equipo del material a ser inspeccionado, y se asignan los papeles.
Preparación	Los miembros del equipo estudian el material individualmente para prepararse para satisfacer los papeles asignados.
Inspección	El equipo realiza una reunión de inspección para encontrar defectos, y registrarlos. El propósito es la detección de los defectos o de violaciones de estándares, alternativas debe ser eliminada por el asesor.
Corrección	El autor revisa el resumen de los defectos detectados, clarificando cuales son realmente defectos y que son mal entendidos en el proceso de la inspección. Entonces, el autor debe modificar para corregir los defectos.
Seguimiento	El asesor o el equipo entero de inspección repasa el producto otra vez, para asegurar que todos los arreglos son eficaces y de que no se ha introducido ningún defecto adicional durante la remodelación.

10

1.3 Análisis

- Heurísticas de verificación de **Requisitos en Alto Nivel** (YUE, 2013; ARORA, 2015)

Description
Describe one action per sentence. A simple sentence must contain only one subject and one verb.
The present simple tense and active form of a verb should be used (user point of view).
Use declarative sentence only. "Is the system idle?" is a non-declarative sentence.
Use words in a consistent way. Keep one term to describe one thing.
Don't use modal verbs (e.g., might, may, should, can, must)
Avoid adverbs (e.g., very).
Don't use negative adverb and adjective (e.g., hardly, never), but it is allowed to use not or no.
Don't use pronouns (e.g. he, this)
Don't use participle phrases as adverbial modifier. For example, the italic-font part of the sentence "ATM is idle, displaying a Welcome message" is a participle phrase.
Use "the system" to refer to the system under design consistently.

1.3 Análisis

- Heurísticas de verificación de **Historias de Usuario** (Lucasen, 2015)

Atomic	A user story expresses a requirement for exactly one feature
Minimal	A user story contains nothing more than role, means and ends
Well-formed	A user story includes at least a role and a means
Conflict-free	A user story should not be inconsistent with any other user story
Conceptually sound	The means expresses a feature and the ends expresses a rationale, not something else
Problem-oriented	A user story only specifies the problem, not the solution to it
Unambiguous	A user story avoids terms or abstractions that may lead to multiple interpretations
Complete	Implementing a set of user stories creates a feature-complete application, no steps are missing
Explicit dependencies	Link all unavoidable, non-obvious dependencies on user stories
Full sentence	A user story is a well-formed full sentence
Independent	The user story is self-contained, avoiding inherent dependencies on other user stories
Scalable	User stories do not denote too coarse-grained requirements that are difficult to plan and prioritize
Uniform	All user stories follow roughly the same template
Unique	Every user story is unique, duplicates are avoided

1.3 Análisis

- Heurísticas de [verificación](#) de **Casos de Uso** (Cockburn, 2000; Leite, 2001; YUE, 2013)

Description
Check that Title defines exactly one situation, i.e., one verb in infinitive (base) form and an object
Check that Goal satisfies exactly one purpose
The subject of a sentence in main and alternative flows should be the system or an actor.
Describe the flow of events sequentially.
Actor-to-actor interactions are not allowed.
Clearly describe the interaction between the system and actors without omitting its sender and receiver.
Check the existence of more than two and less to 10 main flow steps.
Check that every actor participates in at least one step.
Check that every actor mentioned in main flow is included in the actor component.
Check that every resource is used in at least one main flow step.
Check that every resource mentioned in main flow steps is included in the resources component.
Check coherence between the title and the goal.
Ensure that the set of main flow steps satisfies the goal and is within the context.
Ensure that main flow steps contain only actions to be performed.
Check that every use case referenced in main and alternative flows exists within the set of scenarios.
Check that the set of main flow steps of every scenario is not already included in another scenario.
Check that complex alternate/exception (more than 2 steps) is treated by a scenario.

1.3 Análisis

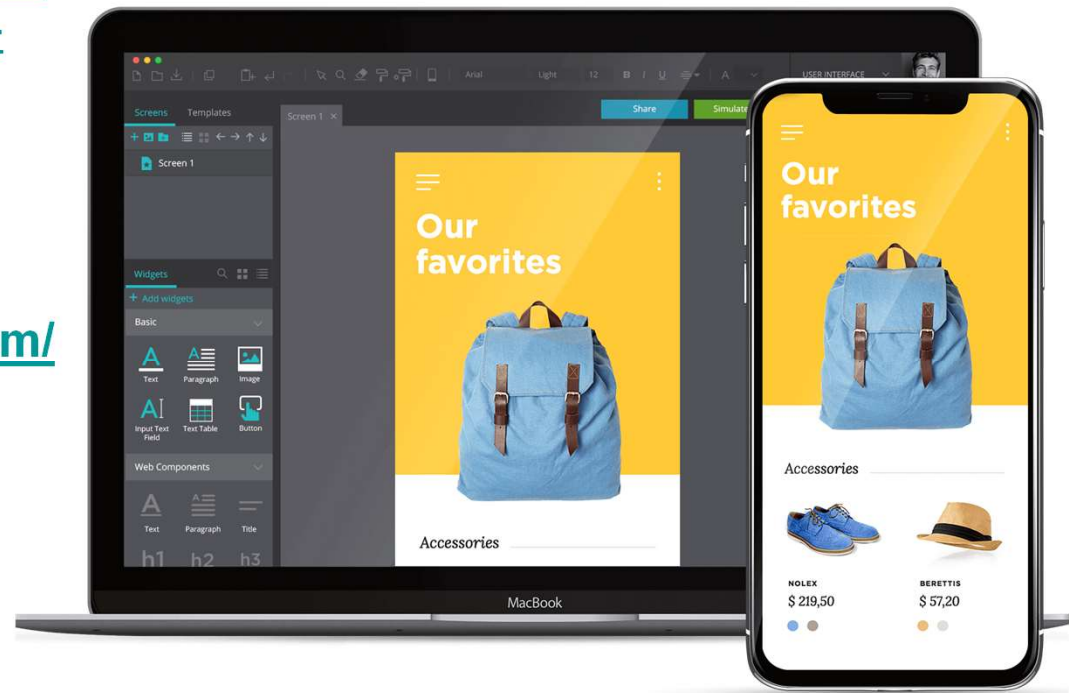
- Heurísticas de verificación de **especificaciones BDD** (Binamungu et al., 2020)

S/n	Quality Aspect
1	A good quality scenario should be concise, testable, understandable, unambiguous, complete, and valuable
2	Reuse of steps across scenarios can improve suite quality
3	Declarative (high level) steps are preferred to imperative (low level) steps
4	Business terminology should be consistently used across the specification
5	Scenarios should focus on the benefit they offer to users, if implemented
6	Scenarios should use the terminology understood by all project stakeholders
7	Each scenario should test one thing
8	Scenario titles should be clear
9	Scenario descriptions should be focused
10	Personal pronoun “I” should be avoided in steps
11	Too obvious and obsolete scenarios should be avoided in the suite
12	Scenario outlines should be used sparingly
13	Scenarios should clearly separate Given, When and Then steps
14	Use past tense for contexts (Given), present tense for events (When), and “should” for outcomes (Then)

1.3 Análisis

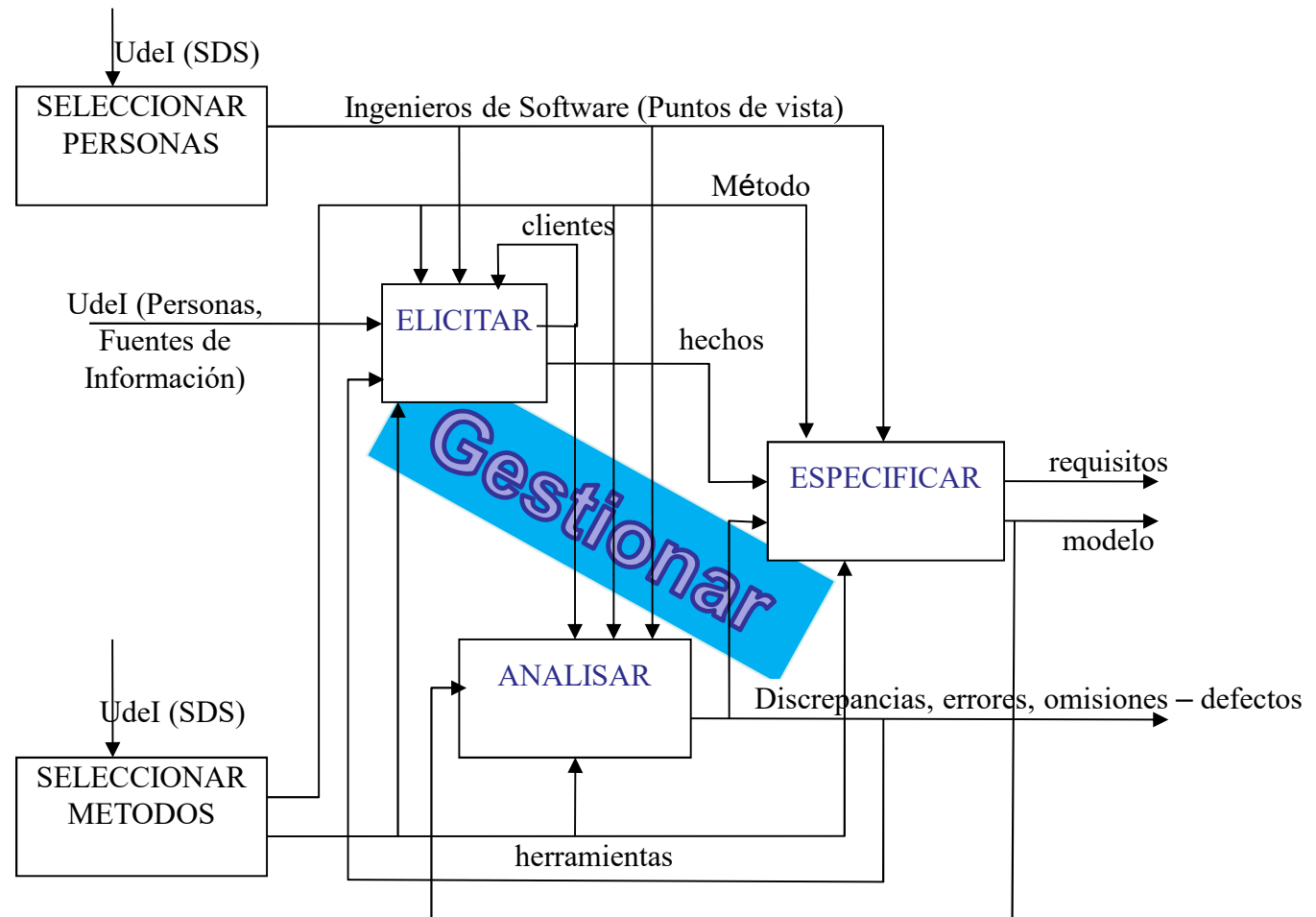
- Validación:
Prototipos

- <https://medium.com/@Vincentxia77/wireframe-vs-mockup-vs-prototype-selection-of-prototyping-tools-febf826cdd98>
- <https://www.justinmind.com/>
- <https://marvelapp.com/>
- <https://proto.io/>



Principales Actividades de la Ingeniería de Requisitos

SADT – Actividades



Julio Leite, 1994

1.4 Gestión

- Gestión de requisitos es transversal al proceso de IR, y comprende las siguientes actividades:
 - Definición de atributos,
 - Visualización,
 - Priorización,
 - Trazabilidad,
 - Control de versión,
 - Configuración y Baseline
 - Control de cambio.

Definición de Atributos

Designación de Atributos para requisitos basado en templates

Attribute Type	Meaning
Identifier	Short, unique identifier of a requirement artifact from the set of all regarded requirements.
Name	Unique, characterizing name.
Description	Briefly describes the content of the requirement.
Version	Current version of the requirement.
Author	Specifies the author of the requirement.
Source	Specifies the source or sources of the requirement.
Stability	Specifies the approximate stability of the requirement. The stability is the amount of changes that are to be expected with regard to the requirement. Possible values can be "fixed", "established", and "volatile".
Risk	Specifies the risk based on an estimate of the amount of damage and loss and the probability of occurrence.
Priority	Specifies the priority of the requirement regarding the chosen prioritization properties, e.g., "importance for market acceptance", "order of implementation", "loss/opportunity cost if not implemented".

Table 8-1 Frequently used attribute types

Definición de Atributos

Attribute Type	Meaning
Person responsible	Specifies the person, group of stakeholders, organization, or organizational unit that is responsible for the content of the requirement.
Requirement type	Specifies the type of requirement (e.g., "functional requirement", "quality requirement", or "constraint") depending on the applied classification scheme.
Status regarding the content	Specifies the current status of the content of the requirement, e.g., "idea", "concept", "detailed content".
Status regarding the validation	Specifies the current status of the validation, e.g., "unvalidated", "erroneous", "in correction".
Status regarding the agreement	Specifies the current status of the negotiation, e.g., "not negotiated", "negotiated", "conflicting".
Effort	Estimated/actual effort to implement the requirement
Release	The designation of the release in which the requirement shall be implemented.
Legal obligation	Specifies the degree of legal obligation of the requirement.
Cross references	Specifies relations to other requirements, for example, if it is known that the implementation of the requirement requires prior implementation of another requirement.
General information	In this attribute, arbitrary information that is considered relevant can be documented, for example, if the negotiation of this requirement is scheduled for discussion during the next meeting with the stakeholders.

Table 8-2 Additional types of requirement attributes

Definición de Atributos

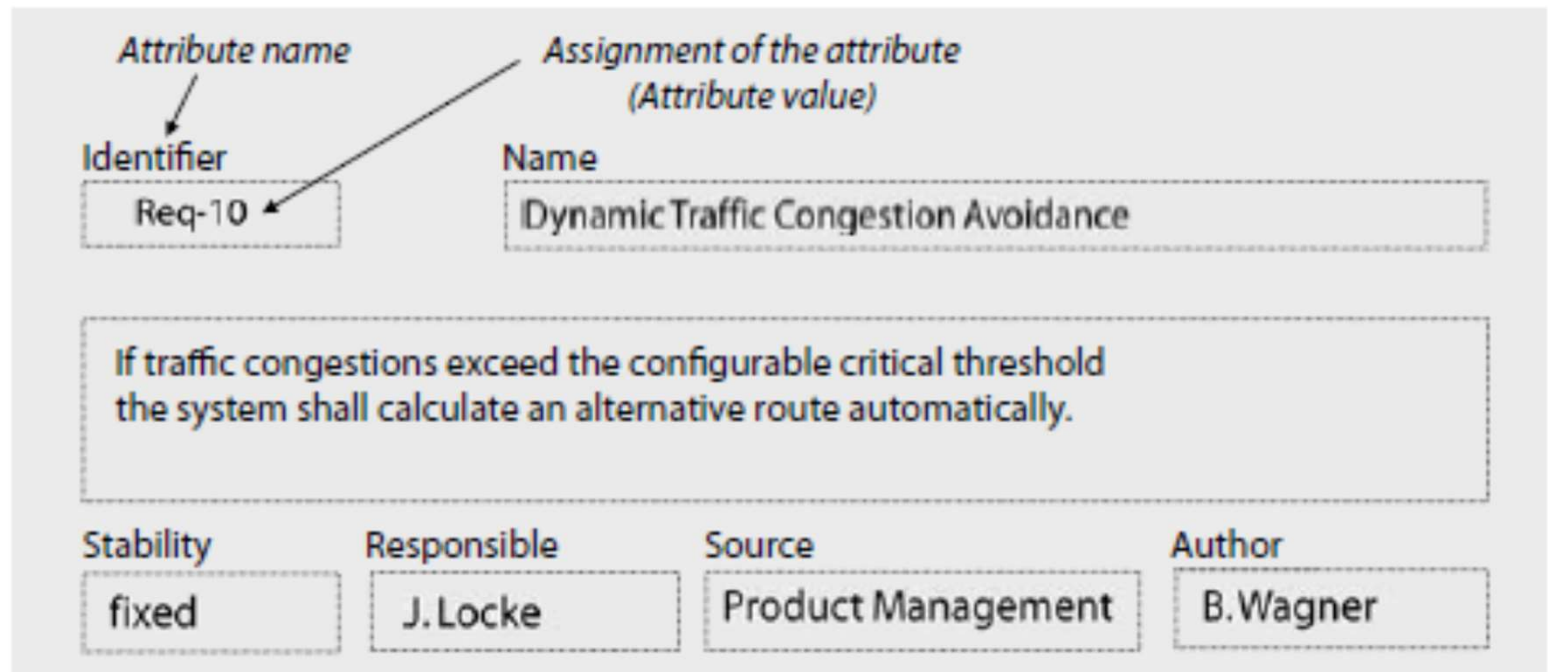


Figure 8-1 Example of requirement attribute assignment

Definición de Atributos

Titulo/Nonmbre	Mover jugador
Objetivo/Descripción	El usuario desea mover al jugador
Pre-condición	Partida iniciada
Pos-condición	
Actor	Usuario
<i>Recursos</i>	<i>Pantalla</i>
Episodios/Flujo Principal	<p>1 El usuario solicita realizar un movimiento.</p> <p>2 El sistema comprueba que el movimiento es válido y lo realiza.</p> <p>3 El sistema muestra la pantalla de juego con la posición actual del jugador y las cajas</p> <p>4 El sistema comprueba si el usuario ha completado el nivel, muestra la pantalla de juego y espera un nuevo movimiento</p>
Alternativas	<p>2.1 Si el movimiento no es válido, el sistema no hace nada y espera un nuevo movimiento.</p> <p>4.1 Si el usuario ha completado el nivel, el sistema carga el siguiente nivel, muestra la pantalla de juego, y espera un nuevo movimiento.</p>

Visualización de Requisitos

Definición de visualizaciones a partir de roles/papeles específicos

Visualizaciones selectivas:

- Seleccionar requisitos
- Ocultar atributos
- Combinar ambos principios

Visualizaciones consolidadas:

- Datos generados o consolidados que no están en descritos en los atributos de los requisitos

Visualizaciones selectivas:

Requirements Basis

Identifier	Name	Description	Author	Source	Responsible	Stability	Status Content	Status Validation	X-Ref
Req-1	"Keyboard Input ..."	"The System ..."	J. Locke	PM	P. Wagner	fixed	concept	unvalidated	Req-3; Req-9
Req-2	"Voice Input ..."	"The System ..."	E. Kurt	PM	P. Wagner	established	concept	in validation	Req-5; Req-123
Req-3	"Reception of ..."	"The System ..."	H. Escher	Maintenance	P. Wagner	established	idea	unvalidated	Req-4; Req-1
Req-4	"Remote Diagn ..."	"The System ..."	M. Bom	Maintenance	M. Bom	volatile	idea	unvalidated	Req-47
Req-5	"Input of miscel ..."	"The System ..."	H. Miller	F. Goldstein	H. Miller	fixed	concept	in validation	Req-33
Req-6	"Accessing reco ..."	"The System ..."	J. Locke	F. Goldstein	M. Bom	fixed	detailed content	validated	Req-45; Req-11
Req-7	"Automatic ..."	"The System ..."	M. Bom	H. Licht	M. Bom	fixed	detailed content	in correction	Req-11
Req-8	"Display of ..."	"The System ..."	H. Miller	J. Locke	M. Bom	volatile	idea	unvalidated	Req-11
Req-9	"Input of miscel ..."	"The System ..."	J. Locke	J. Locke	P. Wagner	volatile	concept	in validation	Req-49
Req-10	"Dynamic ..."	"The System ..."	M. Bom	Customer	P. Wagner	established	detailed content	erroneous	Req-51; Req-9
Req-11	"Voice Control ..."	"The System ..."	H. Miller	Customer	P. Wagner	established	concept	validated	Req-7; Req-81; Req-6
...

1

Views

Identifier	Name	Description	Author
Req-2	"Voice Input ..."	"The System ..."	E. Kurt
Req-3	"Reception of ..."	"The System ..."	H. Escher
Req-10	"Dynamic ..."	"The System ..."	M. Bom
Req-11	"Voice Control ..."	"The System ..."	H. Miller

Selection: Requirements that "J. Locke" is responsible for and have a stability of "fixed."

Display: "Identifier", "Name", "Description", "Author"

Identifier	Name	Author
Req-1	"Keyboard Input ..."	J. Locke
...

Selection: Requirements that are not validated and originate from the source "Product Management" (PM).

Display: "Identifier", "Name", "Author"

Identifier	Name	Description	Author	Source
Req-6	"Accessing reco ..."	"The System ..."	J. Locke	F. Goldstein
Req-7	"Automatic ..."	"The System ..."	M. Bom	H. Licht
Req-8	"Display of ..."	"The System ..."	H. Miller	J. Locke

Selection: Requirements that have a cross reference to requirement R-11.

Display: "Identifier", "Author", "Source", "Responsible", "Stability"

Figure 8-2 Selective views on the requirements

Visualizaciones consolidadas:

Requirements Basis

Identifier	Name	Description	Author	Source	Responsible	Stability	Status Content	Status Validation	X-Ref
Req-1	Keyboard Input ...	The System ...	J. Locke	PM	P. Wagner	fixed	concept	unvalidated	Req-3; Req-9
Req-2	Voice Input ...	The System ...	E. Kurt	PM	P. Wagner	established	concept	in validation	Req-5; Req-123
Req-3	Reception of ...	The System ...	H. Escher	Maintenance	P. Wagner	established	dea	unvalidated	Req-4; Req-1
Req-4	Remote Diagn ...	The System ...	M. Born	Maintenance	M. Born	volatile	dea	unvalidated	Req-47
Req-5	Input of miscel ...	The System ...	H. Miller	F. Goldstein	H. Miller	fixed	concept	in validation	Req-33
Req-6	Acessing reco ...	The System ...	J. Locke	F. Goldstein	M. Born	fixed	detailed content	validated	Req-45; Req-11
Req-7	Automatic ...	The System ...	M. Born	H. Licht	M. Born	fixed	detailed content	in correction	Req-11
Req-8	Display of ...	The System ...	H. Miller	J. Locke	M. Born	volatile	dea	unvalidated	Req-11
Req-9	Input of miscel ...	The System ...	J. Locke	J. Locke	P. Wagner	volatile	concept	in validation	Req-49
Req-10	Dynamic ...	The System ...	M. Born	Customer	P. Wagner	established	detailed content	erroneous	Req-51; Req-9
Req-11	Voice Control ...	The System ...	H. Miller	Customer	P. Wagner	established	concept	validated	Req-7; Req-81; Req-6
...

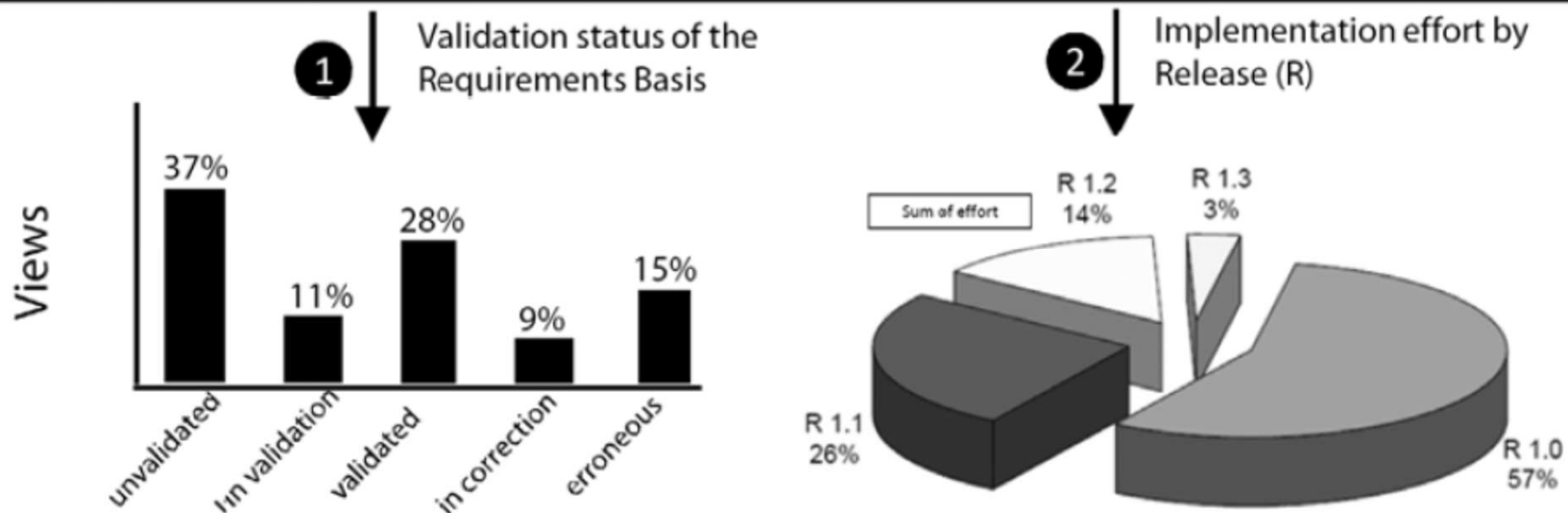


Figure 8-3 Condensed view generated from a requirements basis

Priorización

Todo proyecto que tenga recursos limitados necesita definir prioridades en los requisitos que va a incluir en el producto de software.

Determinar la meta y las restricciones de la priorización;

Determinar los criterios de priorización:

- Costo de implementación,
- Riesgo,
- Costo de una implementación sin éxito,
- Importancia,
- Volatilidad
- Duración de la implementación,
- Precedencia entre requisitos.

Determinar los stakeholders:

Priorización

Técnicas de priorización:

- The spectrum of prioritization techniques spans from simple, single criterion classification to elaborate analytic prioritization approaches, such as [AHP \(Analytical Hierarchy Process\)](#) [Saaty 1980], [Cost-Value-Analysis](#) [Karlsson and Ryan 1997], or [QFD \(Quality Function Deployment\)](#) [Akao 1990].
- **Ranking & Top-Ten:** técnicas consagradas
 - **Ranking:** en esta técnica, los stakeholders seleccionados organizan los requisitos a priorizar con respecto a un criterio específico.
 - **Top-Ten:** en esta técnica, se seleccionan los “n” requisitos más importantes para un criterio definido. Para estos requisitos, un orden de clasificación se determina después. Este orden de “**ranking**” representa la importancia de los requisitos seleccionados con respecto al criterio definido.

Priorización

Relative weight	1 →2 (WeightBenefit)	→1 (WeightDetriment)			→1 (WeightCost)		→0.5 (WeightRisk)			
Requirement	Relative Benefit	Relative Detriment	Total	Value %	Relative Cost	Cost %	Relative Risk	Risk %	Priority	Rank
2 R ₁	5	3	13	16.8	2	13,3	1	9,1	0.941	1
R ₂	9	7	25	32.5	5	33,3	3	27,2	0.692	3
R ₃	5	7	17	22.1	3	20,0	2	18,2	0.759	2
R ₄	2	1	5	6.5	1	6,7	1	9,1	0.577	4
↓ R ₅	4	9	17	22.1	4	26,7	4	36,4	0.489	5
Total	25	27	77	100	15	100	11	100	—	
	3	4	5		6		7		8	9

Figure 8-4 Calculation of priorities in a prioritization matrix according to Wiegers

$$\text{Value\%}(R_i) = \text{Benefit}(R_i) \times \text{WeightBenefit} + \text{Detriment}(R_i) \times \text{WeightDetriment}$$

$$\text{Priority}(R_i) = \text{Value\%}(R_i) / [\text{Cost\%}(R_i) \times \text{WeightCost} + \text{Risk\%}(R_i) \times \text{WeightRisk}]$$

Trazabilidad

Trazabilidad pre-especificación

Trazabilidad post-especificación

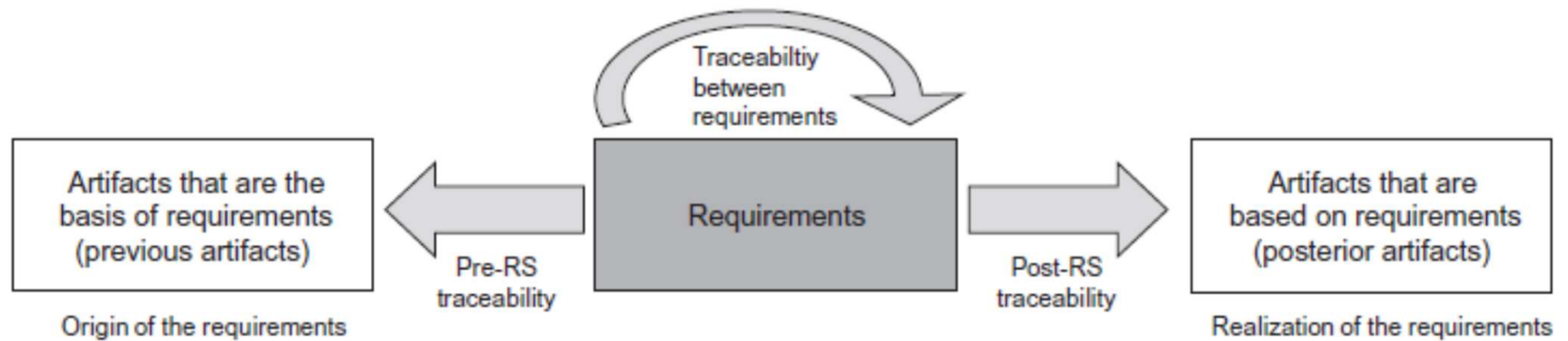


Figure 8-5 Types of requirements traceability

Trazabilidad

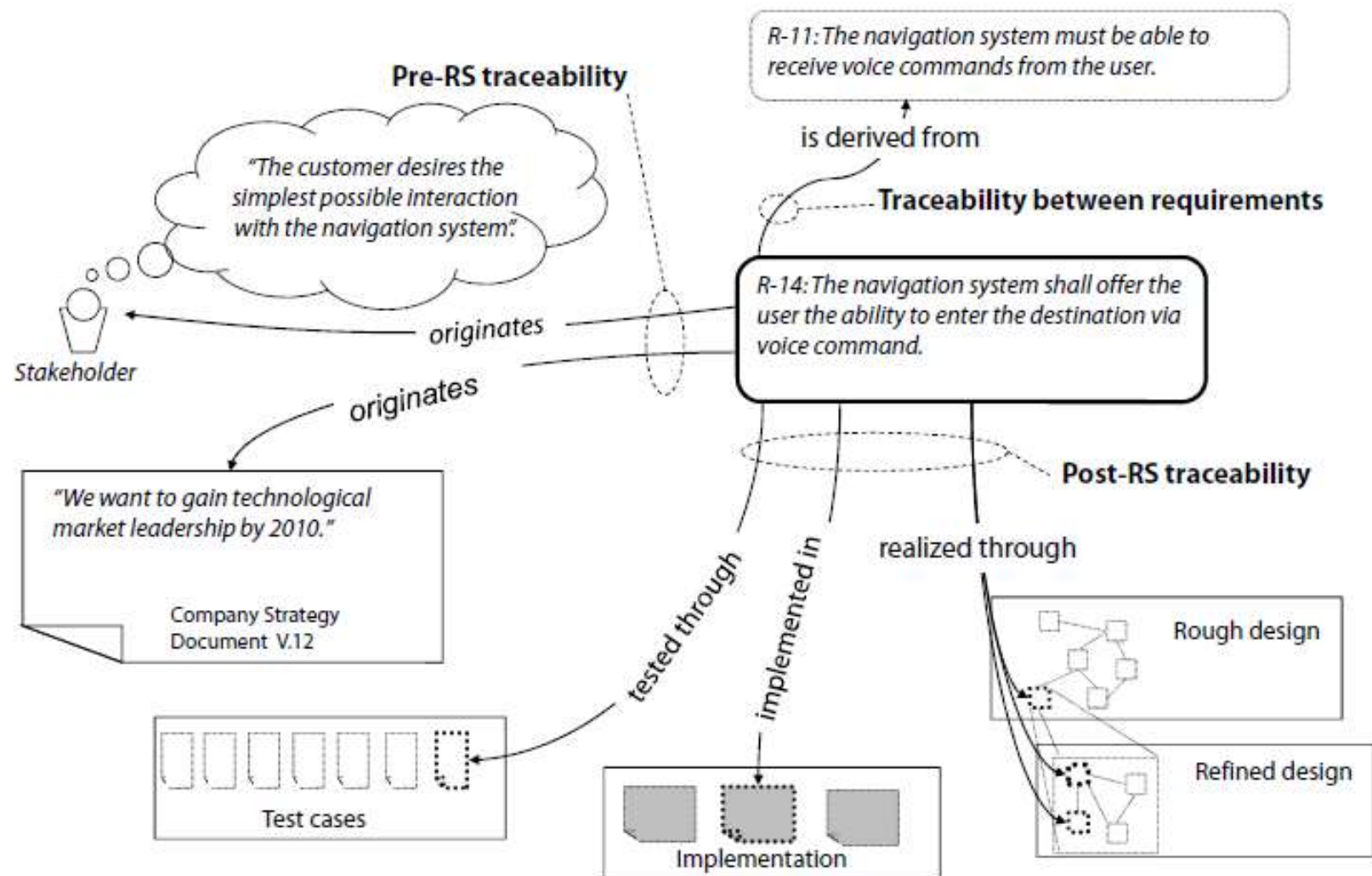


Figure 8-6 Example of the three types of requirements traceability

Trazabilidad

		Target artifacts				
Initial artifacts	<i>derived</i>	Req-1	Req-2	Req-3	Req-4	Req-5
	Req-1		X			
	Req-2			X		
	Req-3					X
	Req-4			X		
	Req-5					

Figure 8-7 Representation of traceability information in a trace matrix

Trazabilidad

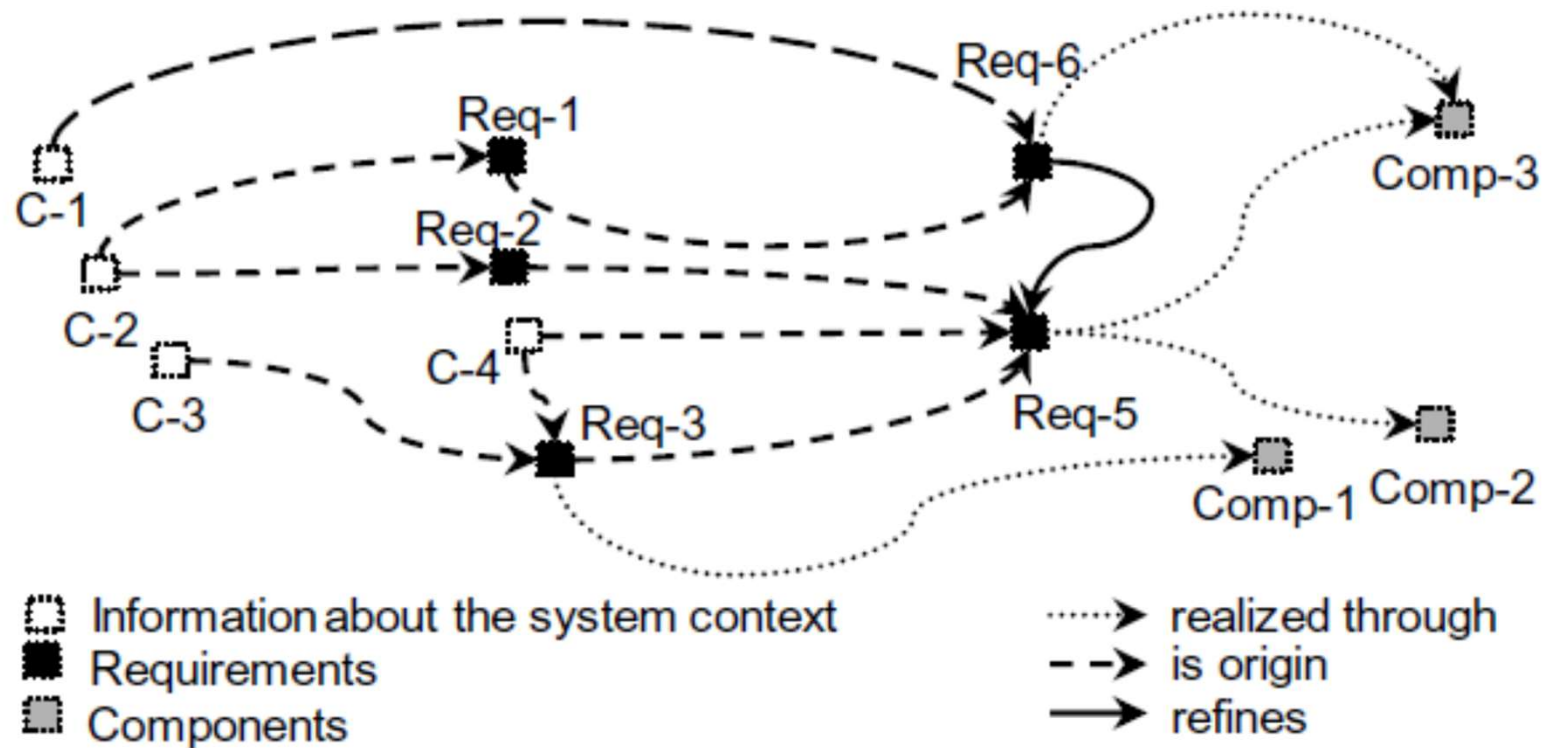


Figure 8-8 Representation of traceability in a trace graph (extract)

Control de Versión

Se puede distinguir la **versión** y el **incremento** de una versión

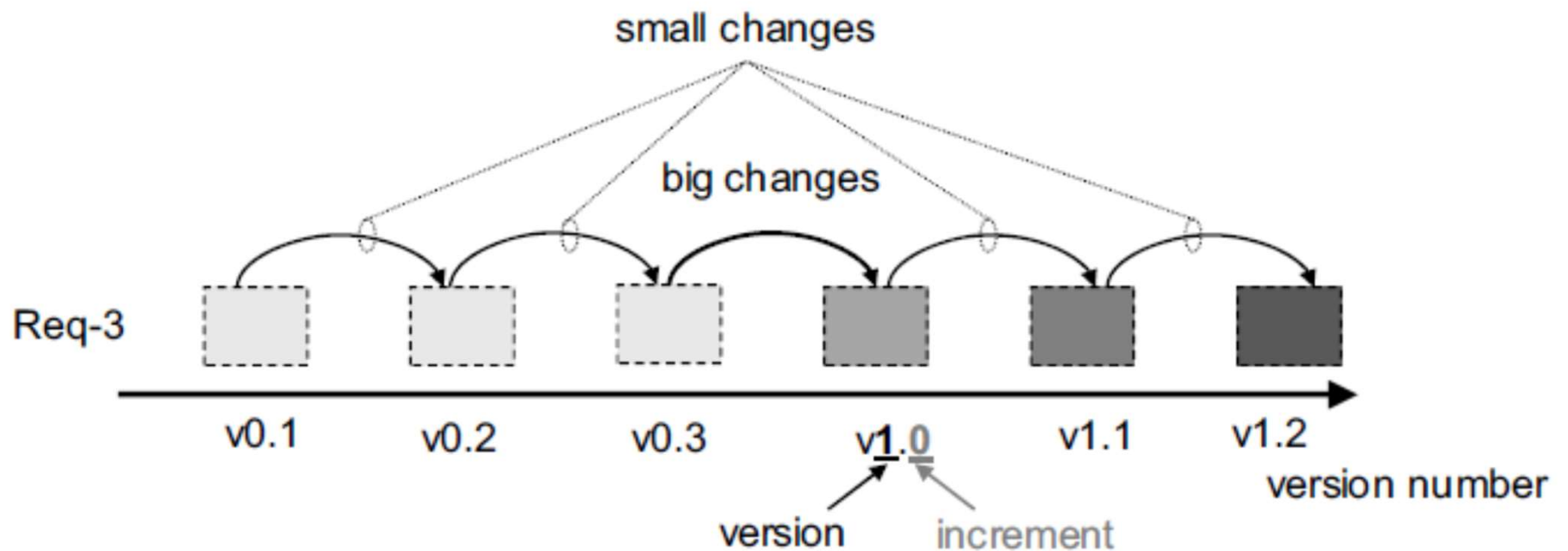


Figure 8-9 *Requirements versions*

Configuración y Baseline

Configuración: conjunto de requisitos donde cada requisito está presente con una versión

Baseline: configuración con versiones estables de requisitos. Sirve para:

- **Planificación** de entregas (release)
- **Estimación** de esfuerzo de implementación
- **Comparación** con productos de la competencia

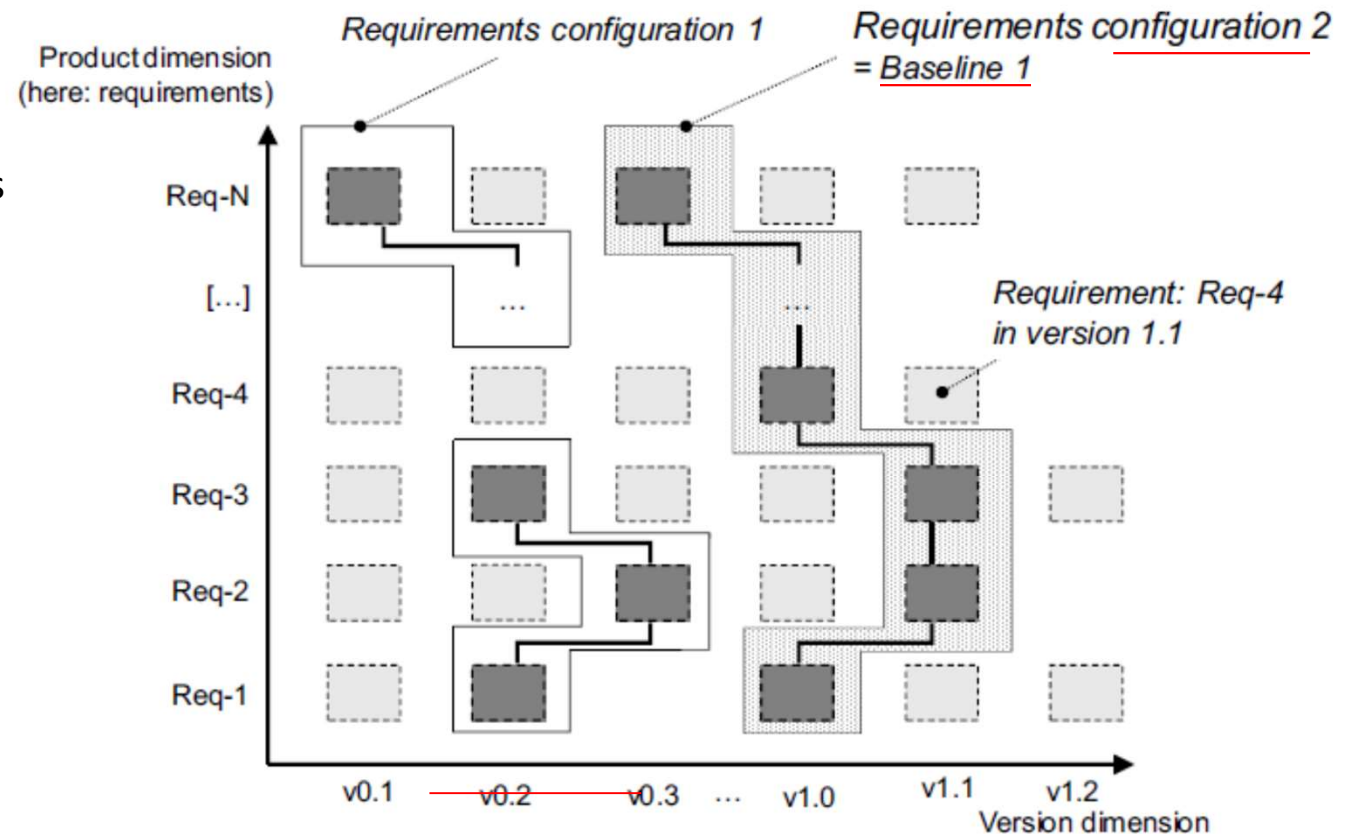


Figure 8-10 Dimensions of configuration management of requirements (based on [Conradi and Westfechtel 1998])

Control de Cambio

Requisitos cambian a lo largo del ciclo de desarrollo de software.

- El **efecto** del cambio propuesto se evalúa a través de la información de trazabilidad y el conocimiento general de los requisitos
- Decidir si un cambio en los requisitos debe ser **aceptado** o **no**.

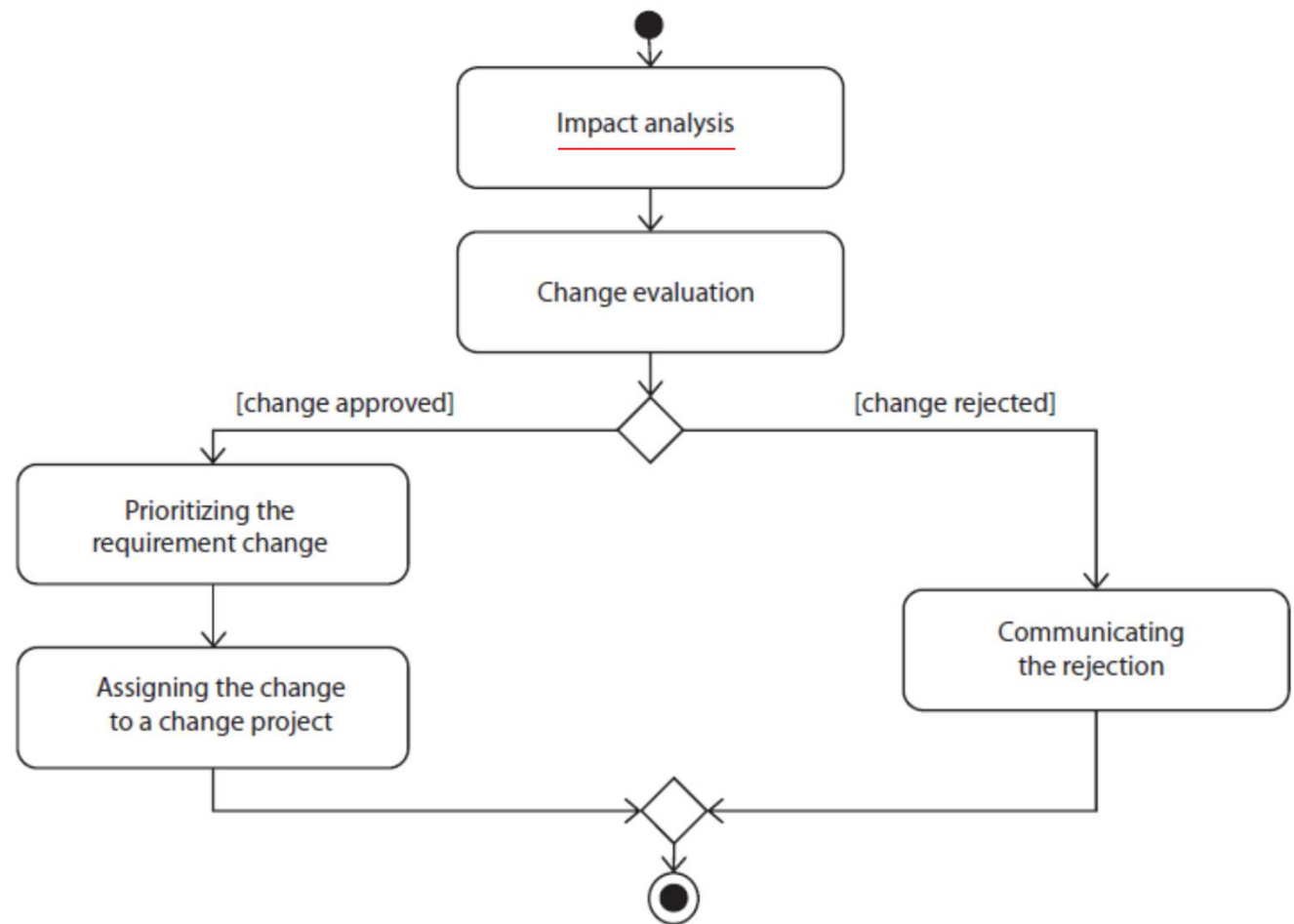


Figure 8-11 Method for handling change requests

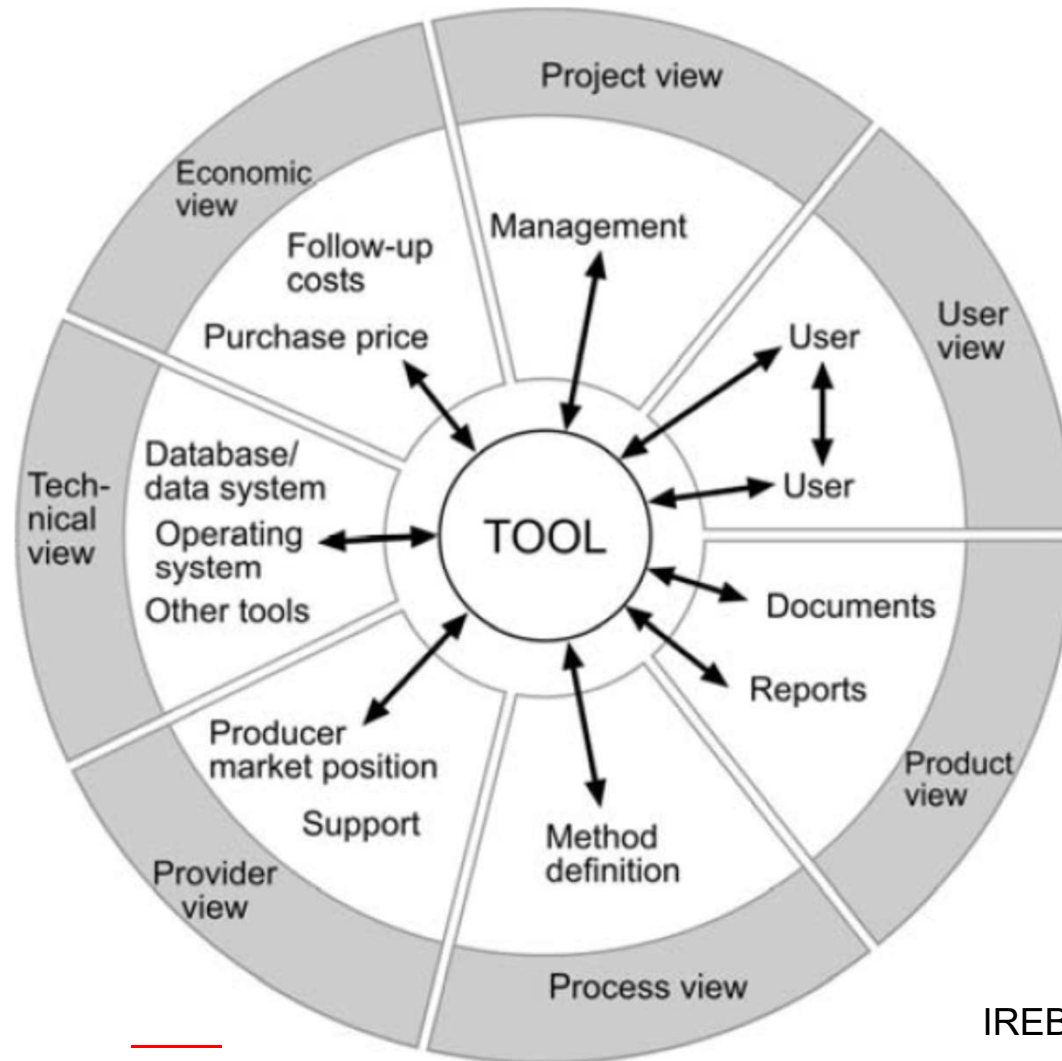
HERRAMIENTAS

3

Herramientas

- Las herramientas de Gestión de Requisitos se caracterizan por tener las siguientes propiedades:
 - Gestión de requisitos y atributos basados en los modelos de información (p.e. template)
 - Organización de requisitos
 - Configuración y gestión de versión en los requisitos
 - Definición de baseline de los requisitos
 - Acceso y gestión multiusuario
 - Gestión de la trazabilidad
 - Consolidación de los requisitos obtenidos
 - Gestión de cambios
 - Análisis de impacto
- Para facilitarnos la difícil tarea de seleccionar una herramienta de gestión de requisitos adecuada, podemos hacer uso de la norma ISO 24766 (Information Technology – Guide for requirements tool capabilities), la cual proporciona una orientación sobre las capacidades deseables que debería aportar una herramienta de Ingeniería de Requisitos. IREB, 2015

Herramientas



IREB, 2015

Figure 9-1 Views on a requirements engineering tool

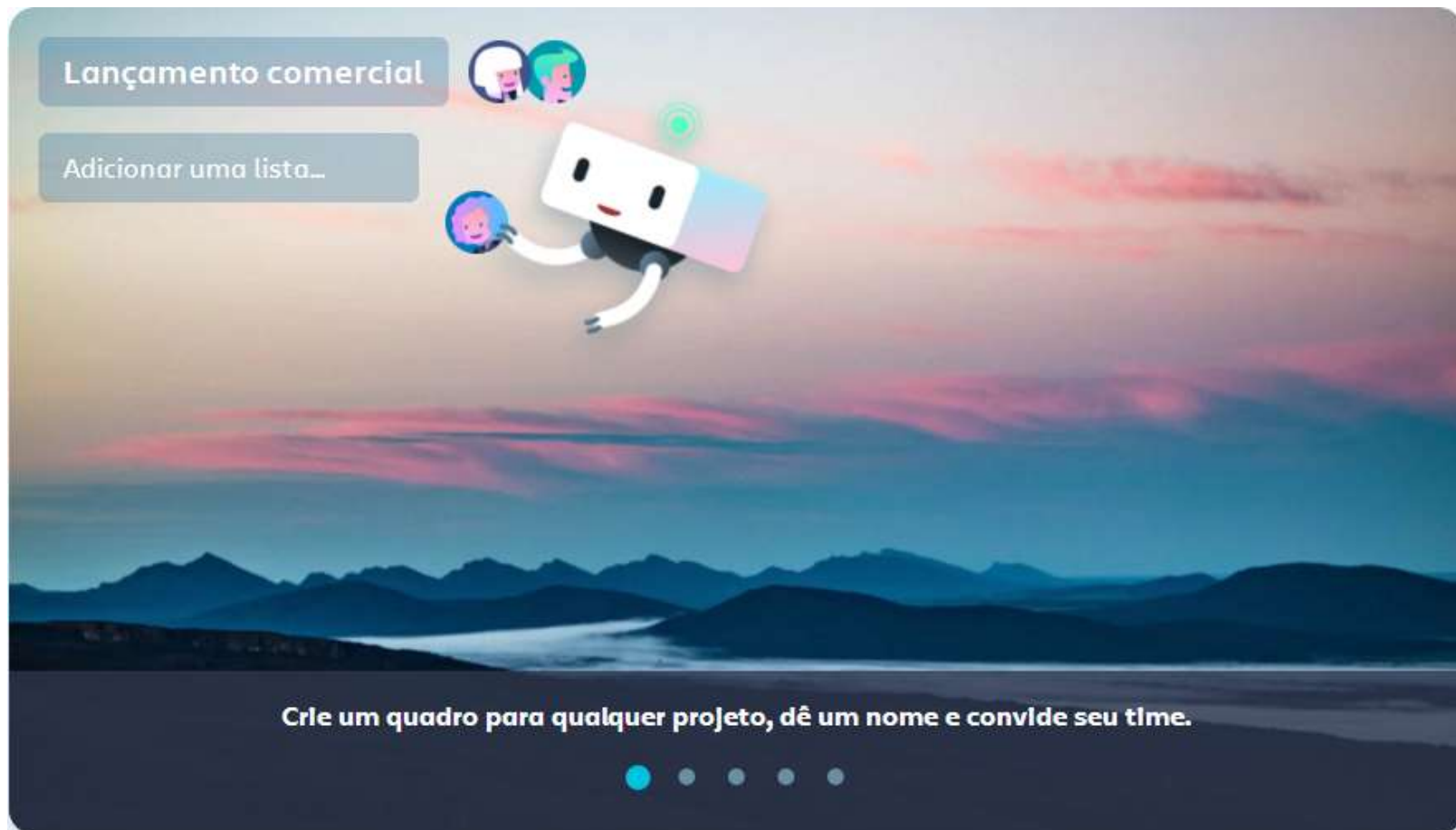
Herramientas



IBM Rational
DOORS Next Generation



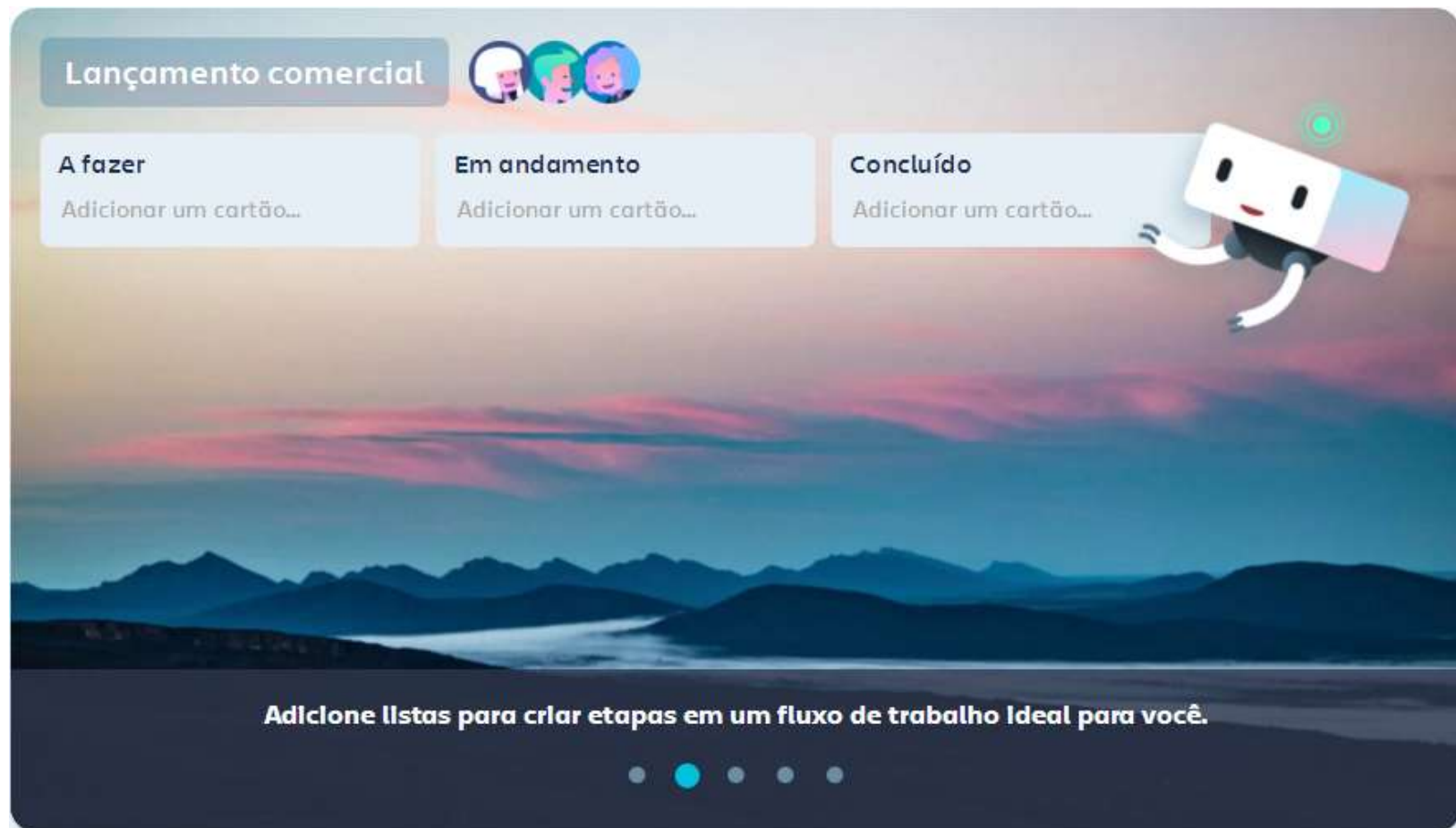
Trello



<https://trello.com>

39

Trello



<https://trello.com>

40

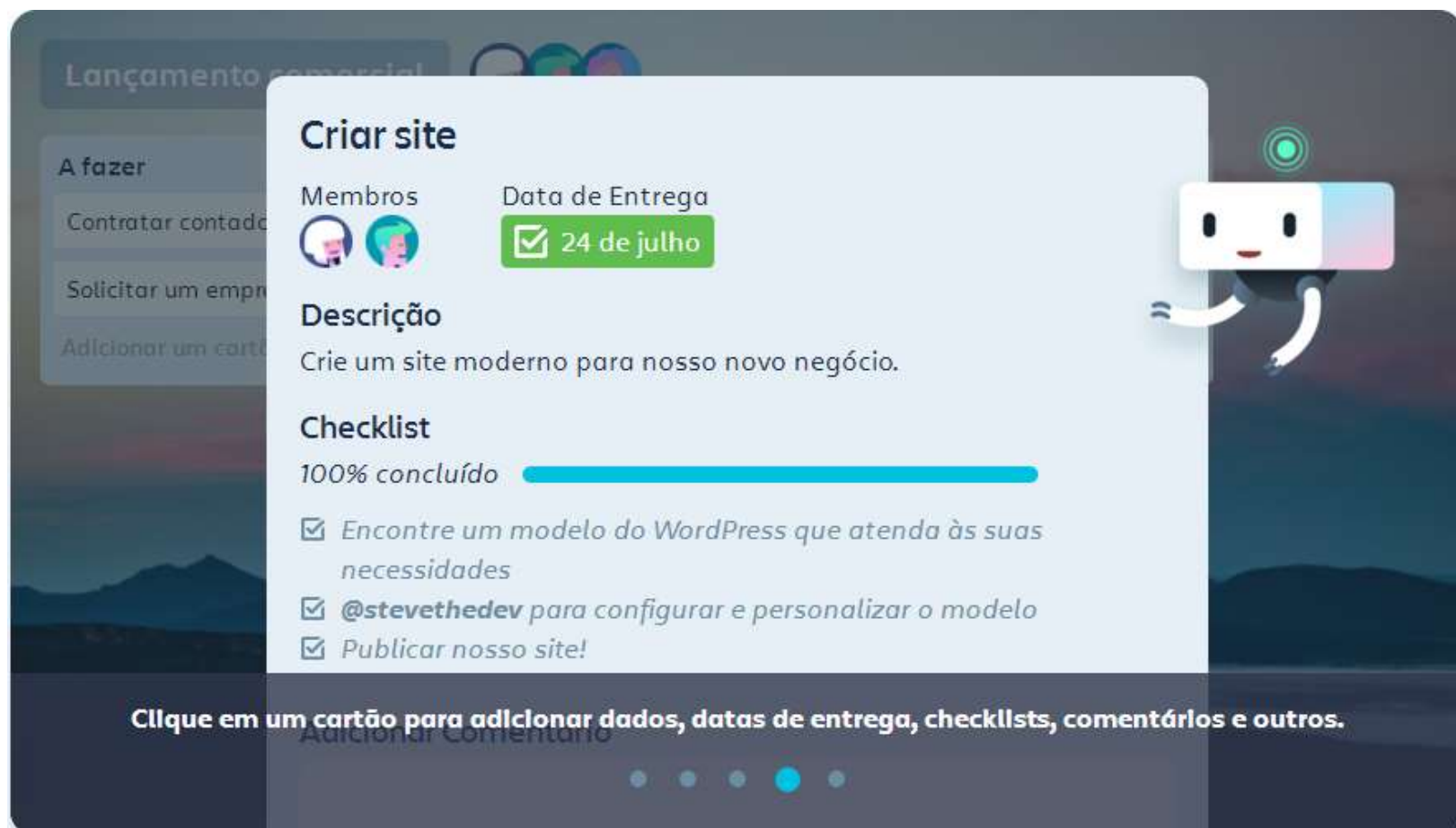
Trello



<https://trello.com>

41

Trello



<https://trello.com>

42

Trello

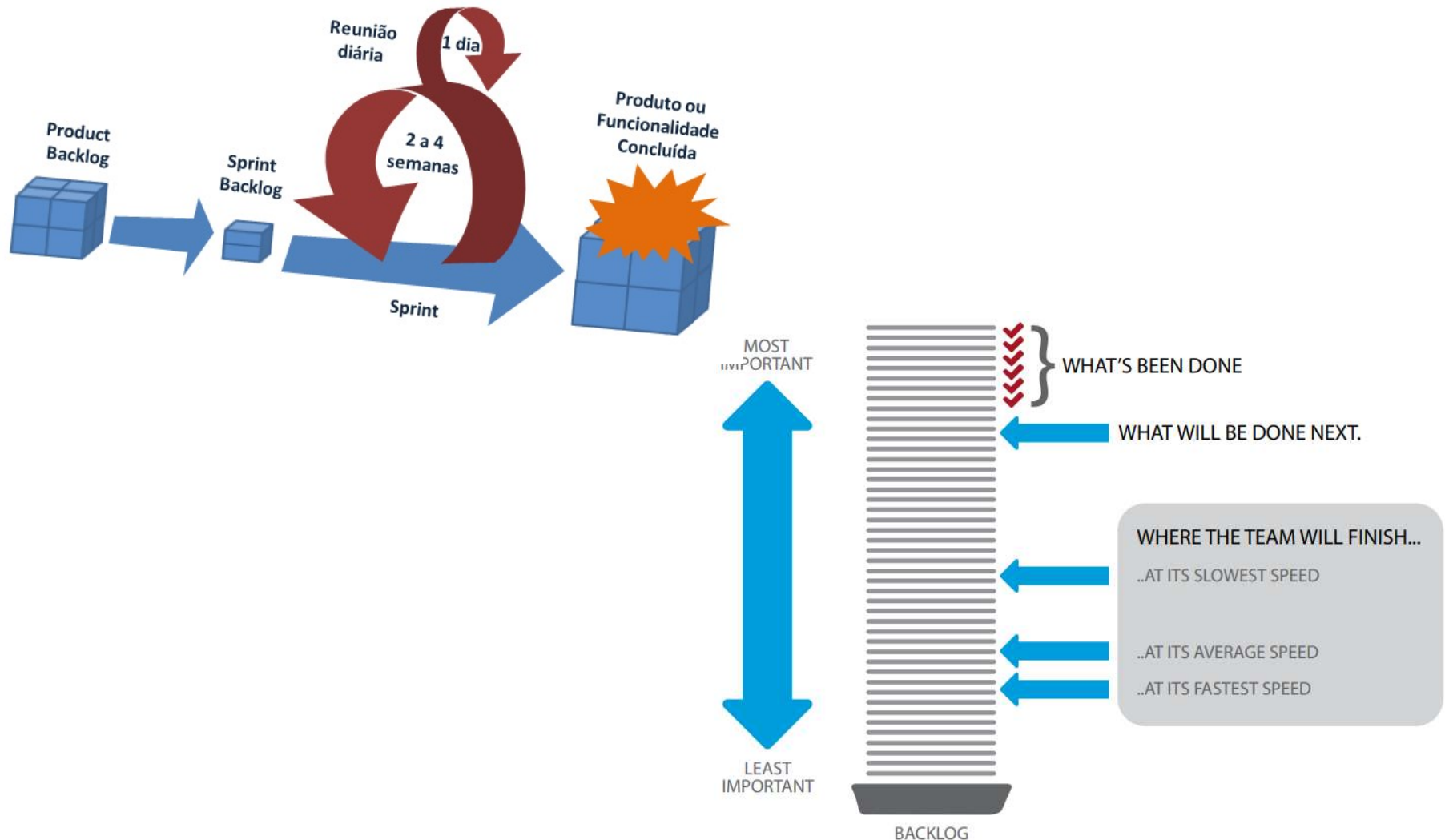


<https://trello.com>

Gestión de Proyectos por Requisitos

Processos Iterativos e Incrementais

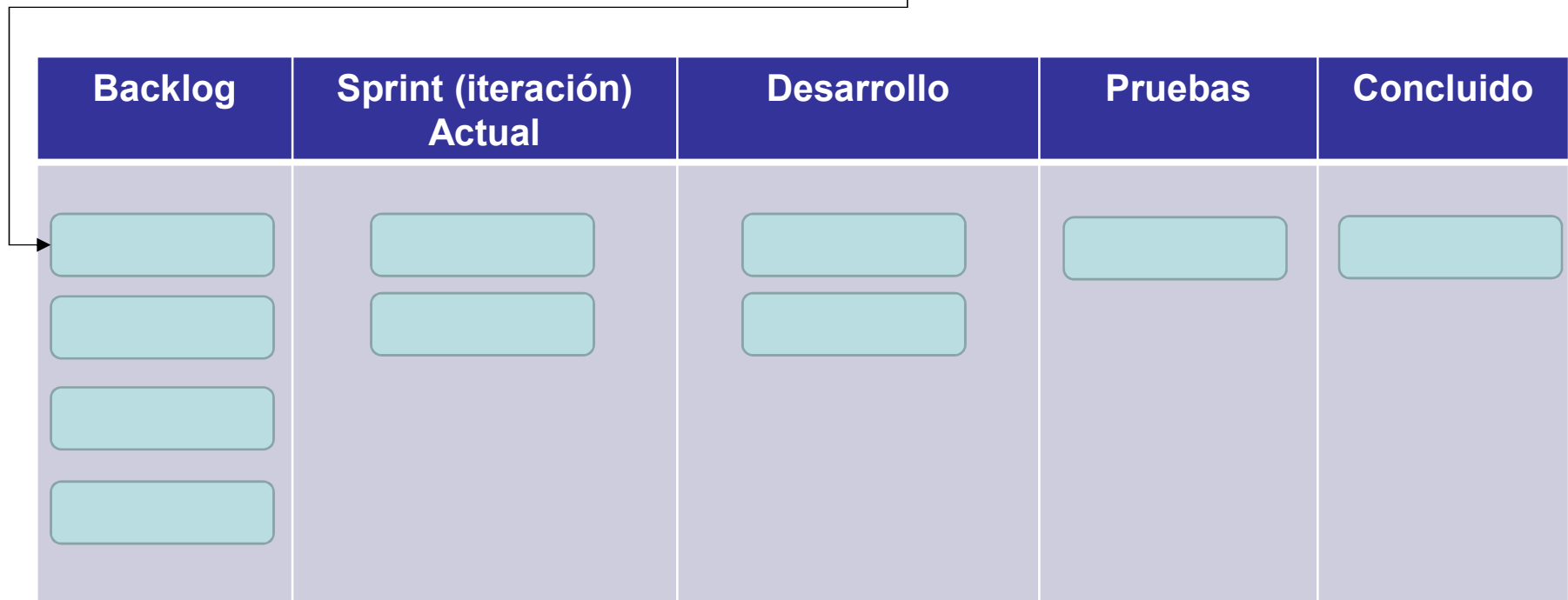
- Requisitos:** Caso de Uso, Funcionalidade/Feature, Escenário, História,



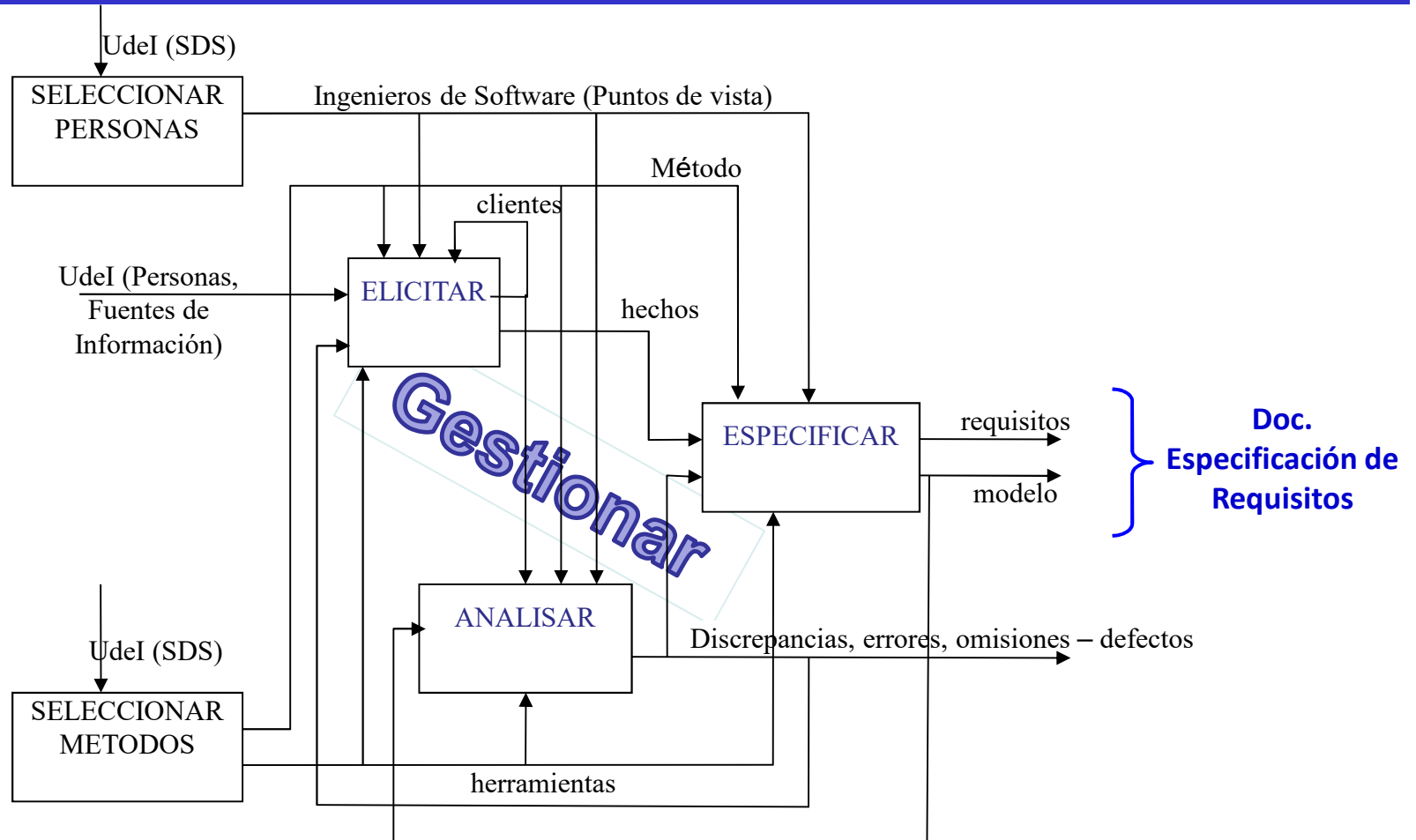
Kanban

- Proyecto

Requisito: Caso de Uso,
Funcionalidad/Feature, Escenario, Historia,
...



Conclusión



Referencias

- **Basado en:**

- Leite, J.C.S.P. 2007. Livro Vivo : Engenharia de Requisitos, <http://livrodeengenhariaderequisitos.blogspot.com/>
- Pohl, K. and Rupp, C. 2015. **Requirements Engineering Fundamentals**. IREB
- **Software Requirements - K Wiegers - 3Ed - Microsoft – 2013.** <http://wdz.eng.br/LivroRequirements3Ed.pdf>
- Jacobson, I., Spence, I., & Bittner, K. (2011). **Use Case 2.0: The Guide to Succeeding with Use Cases**. Ivar Jacobson International. https://www.ivarjacobson.com/sites/default/files/field_iji_file/article/use-case_2_0_jan11.pdf
- Rosana T. Vaccare Braga. 2017. Requisitos de Software. https://edisciplinas.usp.br/pluginfile.php/3142953/mod_resource/content/2/Aula09-Requisitos.pdf
- Elisa Yumi Nakagawa. ENGENHARIA DE REQUISITOS. https://edisciplinas.usp.br/pluginfile.php/58062/mod_resource/content/1/Aula08_Engenharia_Requisitos.pdf
- DO PRADO LEITE, Julio Cesar Sampaio et al. A scenario construction process. **Requirements Engineering**, v. 5, n. 1, p. 38-61, 2000.
- MAVIN, Alistair et al. Easy approach to requirements syntax (EARS). In: **2009 17th IEEE International Requirements Engineering Conference**. IEEE, 2009. p. 317-322.
- Un ejemplo de casos de uso. Sokobanhttp://www.lsi.us.es/~javierj/cursos_ficheros/03.%20Sokoban.%20Un%20ejemplo%20de%20plantillas.pdf