

Metodologías Ágiles: XP, Scrum, Lean

Alejandra Garrido

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

1

Diferencias principales

Waterfall Approach	Agile Approach
Dividido en fases/etapas	Iterativo e incremental
Producto funcionando en fase final	Producto funcionando en cada “build”
Distintas personas en cada fase – pérdida de contexto	Mismo grupo de personas – el contexto se mantiene
Comunicación pobre	Comunicación de calidad
Planeamiento predictivo	Planeamiento adaptativo
Cambios se aceptan solo en algunas etapas	Cambios bienvenidos “changes embraced”



Alejandra Garrido - LIFA - Fac. de Informática - UNLP

3

eXtreme Programming



- Una filosofía de desarrollo de software en base a ciertos valores, principios y prácticas
- Desarrollada por Kent Beck en 1996 (en colaboración con Ward Cunningham)
- Una buena interacción social en el equipo de desarrollo es tan necesaria como las aptitudes técnicas

Alejandra Garido - LIFA - Fac. de Informática - UNLP

4

What is XP?

"XP is my attempt to reconcile humanity and productivity in my own practice of software development

I had begun to notice that the more humanely I treated myself and others, the more productive we all became.

The key to success lies not in self mortification but in acceptance that we are people in a person-to-person business."

"You can't control others' expectations. You can tell them what you know about the Project so their expectations have a chance of matching reality."

Kent Beck. Extreme Programming Explained

Alejandra Garido - LIFA - Fac. de Informática - UNLP

Values, Principles and Practices



- Valores: criterios a gran escala que nos permiten evaluar o juzgar, y le dan sentido a las prácticas. Ej: comunicación.
- Prácticas: lo que hacemos día a día. Son claras y objetivas. Ej: test-first programming
- Principios: unen los valores a las prácticas y les dan contexto. Ej: mutual benefit.

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Valores de XP

- **Comunicación**
 - importante para crear la conciencia de equipo y cooperación efectiva
- **Simplicidad**
 - sistema suficientemente simple para resolver correctamente los problemas de hoy
- **Feedback**
 - no podemos predecir → usamos feedback para acercarnos más y más a la mejor solución
- **Coraje**
 - Acción efectiva frente al miedo
 - Coraje para enfrentar un problema y solucionarlo, teniendo en cuenta las consecuencias para el equipo.
- **Respeto**
 - Cada miembro del equipo es tan importante como todos los otros: para que el desarrollo de software sea humanitario y productivo, es necesario respetar lo que cada miembro del equipo aporta

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Principios

- Humanity (Proceso que tenga en cuenta las necesidades de las personas involucradas)
- Economics (lo que se construye tiene que tener valor para el negocio; tiene más valor cuánto antes gane dinero)
- Mutual Benefit (importancia de las relaciones interpersonales)
- Quality (nunca sacrificarla; lleva a ser más productivo)
- Improvement (hacer lo mejor posible para hoy buscando poder mejorarlo mañana)
- Accepted Responsibility (aceptada, no impuesta)
- Reflection (analizar el trabajo hecho y aprender de ello)

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Otros principios

- Self-similarity (buscar estructuras que funcionen y reusarlas)
- Diversity (dos ideas distintas presentan una oportunidad, no un problema)
- Flow (entregar un flujo constante de software con valor)
- Opportunity (ver a los problemas como oportunidades para aprender)
- Redundancy (resolver los problemas difíciles de más de una manera)
- Baby Steps (para hacer cambios)

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

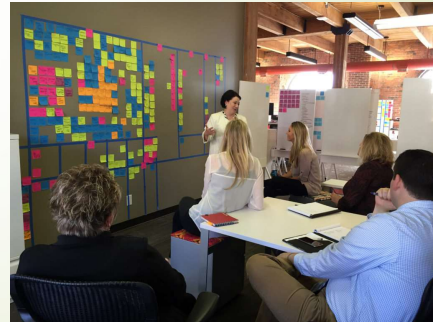
9

Práctica: Sit together

- Una misma oficina para todo el equipo
- Los problemas siempre están relacionados con las personas
- "Verse" hace al proyecto más humano y productivo



Alejandra Garrido - LIFA - Fac. de Informática - UNLP



10

Práctica: Whole team

- We belong
- We are in this together
- We support each others' work, growth and learning
- Equipos cross-funcionales
- Fraccionar el tiempo no funciona

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Práctica: Stories

"Units of customer-visible functionality"

- Escribir y estimar
- Estimar lo antes posible es **clave**
 - diferencia con otras metodologías
 - agrega valor temprano en el proyecto
 - permite tomar decisiones informadas

168 Search by Name

As a help desk operator I want to search for my customers by their first and last names so that customer response times remain short

Story ID:	Story Title:	Importance:
User Story:		Estimate:
As a: <role>		
I want: <some goal>		
So that: <some reason>		
Acceptance Criteria		Type:
And I know I am done when:		<input type="checkbox"/> Search
		<input type="checkbox"/> Workflow
		<input type="checkbox"/> Manage Data
		<input type="checkbox"/> Payment
		<input type="checkbox"/> Report/ View

Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Práctica: Informative Workspace

- Story cards en la pared, en columnas:
 - To do
 - To be estimated
 - In progress
 - Testing
 - Done
 - Future

Kanban



Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

13

Práctica: Pair Programming

- Programar, brainstorm, clarificar, hacerse responsables juntos
- Tiempo en privado
- Rotación frecuente



Alejandra Garrido - LIFA - Fac. de Informática - UNLP

14

Práctica: Test-First Programming

- Escribir el test **antes** del código
 - ayuda a enfocarse en lo que hay que resolver
 - si el test es difícil de escribir, no es un problema de testing, es un problema de diseño
 - da credibilidad y confianza
 - ritmo: test – code – refactor, test – code – refactor

“Continuous testing reduces the time to fix errors by reducing the time to discover them”

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Práctica: Continuous Integration

"Team programming isn't a divide and conquer problem.
It's a divide, conquer and integrate problem"

- Integrar y testear los cambios cada 2 hrs máximo

Práctica: Ten-minute Build

"Automatically build the whole system and run all of the
tests in 10 minutes"

- Más largo no se usa, no se aprovecha el feedback
- Más corto no da tiempo para el café

Práctica: Incremental Design

"The most effective time to design is in the light of experience"

- Invertir en el diseño cada día, y esforzarse en el mejor diseño que se ajuste a las necesidades de hoy
- Diseñar todo al principio surge de la premisa de que el costo de solucionar defectos crece exponencialmente con el tiempo, pero entonces con el diseño pasa lo mismo!

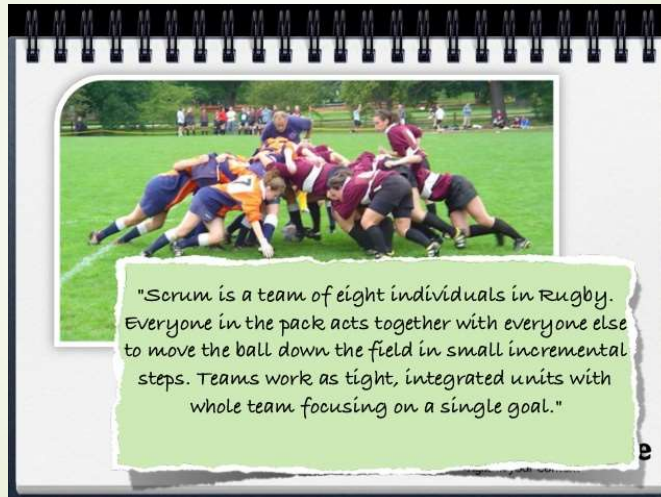
Alejandra Garrido - LIFA - Fac. de Informática - UNLP

XP se diferencia en que:

- Ciclos de desarrollo cortos
- Planificación incremental
- Schedule flexible
- Confianza en:
 - tests automatizados
 - comunicación oral, tests y código fuente para comunicar
 - evolución del diseño durante todo el ciclo de vida
 - colaboración mutua
 - prácticas que se ajustan al equipo, a los instintos de corto plazo y a los intereses a largo plazo del proyecto

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Scrum



Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Scrum - historia

- Artículo de 1986 Takeuchi and Nonaka, publicado en Harvard Business Review
 - basado en lo que era el desarrollo de productos en Toyota, Honda, Fujitsu, Cannon, etc.
 - observaron que sus fases de construcción se solapaban, construían grupos interdisciplinarios trabajando en el mismo lugar físico.
- Jeff Sutherland crea el proceso Scrum moderno en 1993 (basándose en el artículo anterior y en otras cosas como "organizational patterns")

Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Scrum - historia

- Ken Schwaber desarrolla un proceso similar en su compañía y lo presenta en un workshop en OOPSLA'95
- Libro de Ken Schwaber y Mike Beedle en 2001: "Agile Software Development with Scrum"
- Mucha literatura en la actualidad. Recomendada:
 - Essential Scrum. A Practical Guide to the Most Popular Agile Process. Kenneth Rubin. 2013

Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Scrum framework

- Es un framework: se puede adaptar, customizar a las necesidades de la empresa
- Está basado en valores, principios y prácticas, que no pueden ignorarse pero pueden adaptarse
- Valores de Scrum:
 - honesty, openness, courage, respect , focus, trust, empowerment, collaboration

Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

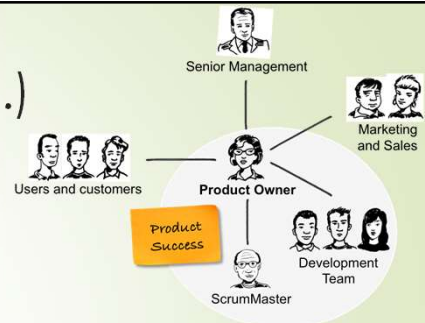
Scrum - características

- Proyecto organizado en un **Product Backlog**
- El trabajo se estructura en iteraciones cortas llamados **sprints**
- Un **equipo** de desarrollo multi-disciplinario, auto-organizado
- Al final de cada sprint el equipo de lograr un **potentially shippable product increment (MVP)**
- Scrum consta de:
 - 3 roles
 - 3 artefactos
 - 3 reuniones

Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

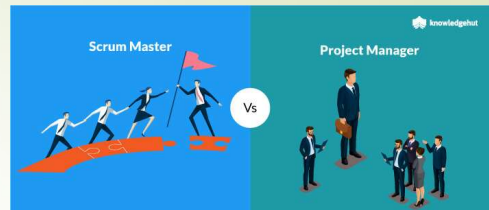
Rol – Product Owner (P.O.)

- Visión del producto (poder de decisión)
- Representa el interés de los stakeholders
- El dueño del product backlog, es decir que tiene la autoridad y la responsabilidad de decidir:
 - qué características y funcionalidad se van a construir (puede agregar o sacar user stories)
 - en qué orden se van a construir (asigna prioridad a las stories, de manera que lo que tenga más valor se construya primero)
- Acepta o rechaza resultados
- Planifica releases



Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Rol – Scrum Master



- Responsable del cumplimiento del proceso
- Evangelizador del grupo y la organización
- Encargado de calzar Scrum en la cultura de la organización (ayuda en los cambios de procesos que puede causar la adopción de Scrum)
- Elimina obstáculos
- ScrumMaster **no es un manager**, sino un facilitador (**coach**) y una barrera contra influencias negativas.
- Responsable por la calidad

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Rol – Development Team

- Grupo de personas responsables de: diseñar, construir y testear el product deseado
- Auto organizado
 - Eligen stories
 - Definen tareas
 - Los integrantes se autoasignan trabajo
- **Multidisciplinario**
- **Colectivamente responsables** por el éxito de cada iteración.



Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Artefacto: Product backlog

- Lista de user stories y features (P.B.I.s)
- Priorizada y estimada
- Preparada por el P.O. al principio del proyecto
- No es fija! Evoluciona constantemente en respuesta al **feedback**
- Estimación: **story points** (horas ideales o días ideales - tamaño relativo de cada PBI comparado con otros)

	Item #	Description	Est	By
Very High				
	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
		- Add licensing	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		- Analysis Manager	-	-
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
		- Enforce unique names	-	-
	7	In main application	24	KH
	8	In import	24	AM
		- Admin Program	-	-
	9	Delete users	4	JM
		- Analysis Manager	-	-
		- When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	-	-
	10	- Query	8	TG
		- Support for wildcards when searching	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
		- Population Genetics	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	Add icons for v1.1 or 2.0	-	-
		- Pedigree Manager	-	-
	20	Validate Derived kindred	4	KH
Medium				
		- Explorer	-	-
		- Launch tab synchronization (only show queries/analyses for logged in users)	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Estimación de PBIs

☐ Story ID:
 Story Title:

User Story:

As a: <role>
 I want: <some goal>
 So that: <some reason>

Importance:

Estimate:

Acceptance Criteria

And I know I am done when:

Type:

☐ Search
 ☐ Workflow
 ☐ Manage Data
 ☐ Payment
 ☐ Report/View

- La estimación la realiza el **equipo** colectivamente
- Debe ser realista, no demasiado precisa (estimar es costoso; "good-enough" alcanza)
- **Story points**: mide la magnitud del PBI en términos de: complejidad, tamaño, esfuerzo. Sirve para comparar entre stories.
- Velocidad del equipo: $\sum_{stories\ done\ in\ a\ sprint} story\ points$
- Tiempo hasta el release: $\sum story\ points / average\ team\ velocity$

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Sprint

- Es cada iteración o ciclo en el que se divide el trabajo
- Suele ser de 2 a 4 semanas. Se fija según el equipo / proyecto
- Al comienzo del sprint, el equipo elige los P.B.I.s que va a desarrollar del product backlog (realísticamente, a una velocidad cómoda)
- Durante un sprint no se aceptan cambios
- Se desarrollan las tareas que componen los PBIs para llegar a completarlas (done)
- **"Done"**: grado de confianza de que el trabajo realizado tiene buena calidad y es "potentially shippable".
 - Como mínimo debería incluir diseño, build, integración, testing y documentación. Pero al ppio podría ser una unidad de funcionalidad suficiente y usable para generar feedback

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Artefacto: Sprint backlog

- User stories seleccionadas en un sprint durante el planeamiento
- Se divide cada story en tareas pequeñas (de menos de 16 hrs) y se estima cada tarea.
- Manejado por el equipo

User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	...
As a member, I can read profiles of other members so that I can find someone to date.	Code the ...	8	4	8	0		
	Design the ...	16	12	10	4		
	Meet with Mary about ...	8	16	16	11		
	Design the UI	12	6	0	0		
	Automate tests ...	4	4	1	0		
	Code the other ...	8	8	8	8		
As a member, I can update my billing information.	Update security tests	6	6	4	0		
	Design a solution to ...	12	6	0	0		
	Write test plan	8	8	4	0		
	Automate tests ...	12	12	10	6		
	Code the ...	8	8	8	4		

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Artefacto: Potentially shippable product increment

- Lo que se produce al final de la ejecución del sprint
- Definición de **Minimum Viable Product (MVP)**: es un incremento del producto que un equipo puede *entregar* o *vender* para obtener el feedback máximo de los clientes o aprender de ellos lo más posible con el menor esfuerzo
- Definición de **Minimum Testable Product (MTP)**: para testear la viabilidad de una idea, el mercado, suposiciones, usabilidad del producto, los posibles usuarios
 - Dirigido a "early adopters"

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Reuniones – Sprint planning

- **Inicio del Sprint** (4 a 8 horas)
- Asisten: el equipo, el Scrum master y el PO
- **Selección de items:**
 - El PO presenta los items de mayor prioridad
 - El equipo estima la dificultad de cada story (tiempo de desarrollo o puntos de dificultad).
 - El equipo acuerda sobre la incorporación de cada tarea.
- **Planeamiento del sprint:**
 - Descomposición en tareas
 - Diseño

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Reuniones - Daily meeting

- 15 min, parados frente al Task board
- No participa el PO
- Cada miembro responde a 3 preguntas
 - Qué hizo ayer?
 - Qué va a hacer hoy?
 - Qué impedimentos existieron?
- Las discusiones finas de diseño se postergan (se resuelve luego entre los interesados)

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Reuniones – Review Meeting

- Final del Sprint (4hs máximo)
- Asisten: el PO, otros stakeholders, clientes, SM y el equipo
- Se enfoca en revisar lo desarrollado durante el sprint, en el contexto del esfuerzo general
- Alguien del equipo presenta el desarrollo del sprint
- Es la oportunidad de que el equipo pueda aprender más sobre el negocio y el marketing del producto a través del feedback sobre lo hecho, y adaptar el producto de manera acorde

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Otras actividades

- **Sprint retrospective:** se reúne el equipo con el SM
 - Es la oportunidad de reflexionar sobre el proceso de desarrollo (inspeccionarlo y adaptarlo)

Alejandra Garrido - LIFA - Fac. de Informática - UNLP

El proceso Scrum



Alejandra Garrido - LIFA - Fac. de Informática - UNLP

Task Board: para nosotros Trello

Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... DC 8 Test the... SC 8 Test the... SC 8 Test the... SC 6
	Code the... 2	Code the... 8	Test the... SC 8		
	Test the... 8	Test the... 4			
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... SC 8 Test the... SC 8 Test the... SC 6
	Code the... 4	Code the... 6			

Alejandra Gamido - URA - Fac. de Informática - UNLP

The screenshot displays the Trello web interface. At the top, there's a navigation bar with 'Boards', 'Personal', and 'Private' tabs. The main area is titled 'Welcome Board' and contains three columns: 'Basics', 'Intermediate', and 'Advanced'. Each column has a list of instructions and tips for using Trello. On the right side, there's a 'Menu' panel with various settings and activity logs. The bottom of the screen shows a URL bar with the address 'https://trello.com/c/t9gDCQPM/16-need-help'.

Trello: listas, tarjetas, etiquetas (para la entrega del 12/9)

- Listas:
 - Product backlog
 - Sprint backlog (todas las user stories para hacer en el sprint)
 - Doing (stories comenzadas del sprint actual)
 - Done (stories terminadas del sprint actual)
 - Sprint n+1 backlog (para ir agregando los items que vayan surgiendo)
- Cada tarjeta:
 - Checklist con tareas
 - Cada tarea con story points asignados o si está en el backlog, prioridad
 - Etiquetas: UX task, Dev task (al menos)

Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Prácticas de nuestra materia

- Diseño centrado en el usuario
- Recorrido Cognitivo
- Refactoring interno y externo
- Testing:
 - unit testing,
 - user testing,
 - A/B testing

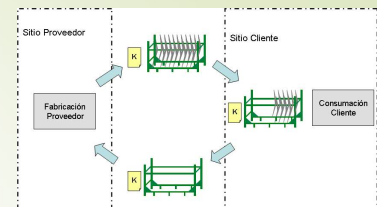
Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Lean Production



- El término "lean" nace en el '88
- Análisis de la industria automotriz japonesa y su producción de autos con:
 - mayor calidad,
 - mayor rapidez de reposición,
 - en la mitad de horas hombre que la industria automotriz norteamericana (Ford)

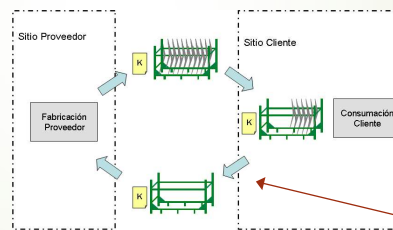
Características Lean



- Tarjetas Kanban: se revierte el flujo de información
- En vez de seguir un plan de producción ("push") para forzar productos al mercado, se producen partes en pequeños lotes según la necesidad del mercado ("**pull**")
- Just-in-time (**JIT**) vs. Just-in-case
- Minimiza el desperdicio (waste) y el stock de inventario, y la producción está guiada por la demanda
- Comienza por definir la cantidad ideal de productos a producir: ni tan grande que haga difícil llegar a producir y almacenar, ni tan pequeña que no llegue a cubrir la demanda

Lean software development

- Waste = burocracia, documentación excesiva, todo lo que no produce valor para el cliente
- Aprendizaje a través de ciclos cortos y builds frecuentes



- Feedback pulling vs. requirements pushing
- TDD: test pulling

Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Kanban. Principios

TO-DO – IN PROGRESS – DONE



- **Visualizar todo lo que está ocurriendo en un momento dado.** Cada elemento y su estado de avance se ve en el contexto de todo el trabajo
- **Limitar la cantidad del trabajo en curso, o Work In Progress (WIP).** Hay que poner un máximo a la cantidad de tareas que se pueden gestionar al mismo tiempo (de forma eficiente y sin desperdicio) y los límites visuales del tablero ayudan a percibir físicamente esa limitación de máximos.
- **Enfocarse en el flujo (continuidad) del trabajo.** En cuanto se termina un elemento, se inicia otra tarea del *backlog*. Cualquier interrupción del flujo presenta una oportunidad de mejora. Para ello, es fundamental que el *backlog* esté correctamente administrado, priorizado y categorizado.
- **Mejora continua.** Requiere monitoreo y análisis constante, aliviando fricciones y obstrucciones en el flujo de producción

Alejandra Garrido - LIFA - Fac. de Informatica - UNLP

Bibliografia

- www.agilemanifesto.org
- "Extreme Programming Explained. Embrace change". Kent Beck. Addison-Wesley. 2004.
- "Agile Software Development with SCRUM". Michael Beedle y Ken Schwaber. Prentice Hall. 2001
- "Essential Scrum: A Practical Guide to the Most Popular Agile Process". Kenneth Rubin. Addison-Wesley. 2013
- "Lean Software Development: a Tutorial". Poppendieck & Cusumano. IEEE Software 2012
- "Lean Software Development: An Agile Toolkit". Poppendieck & Poppendieck. Addison-Wesley 2003