



Въведение в курса

Разработка на софтуер

Съдържание

- Етапи в софтуерната разработка
- Методологии за разработка на софтуер
- Инструменти

Етапи в софтуерната разработка

- Планиране
- Анализ
- Дизайн
- Разработка & имплементация
- Тестване
- Инсталация и поддръжка

Етапи в софтуерната разработка

Планиране

- Етапа включва
 - Събиране на бизнес изисквания
 - Среци между заинтересованите страни
 - Разработчиците трябва добре да се запознаят с изискванията на клиента
 - Предвиждане на евентуалните проблеми, които могат да възникнат

Етапи в софтуерната разработка

Анализ

- Екипът дефинира детайлно целия проект, като проверява неговата приложимост
- Работният процес е разделен на малки задачи, за да може разработчици, тестъри, дизайнери и мениджъри на проекта да дадат оценка за своите задачи
 - Те оценят дали дадена задача е приложима относно цена, време, функционалност, надеждност и др.

Етапи в софтуерната разработка

Дизайн

- Дефинира се дизайнът на софтуера на база събраните изисквания от фазата “Планиране”
- Определяне на системните и хардуерните изисквания за проекта
- Създаване на стратегия за тестване
 - Кое да се тества?
 - Как да се тества?
- Съдържа цялостна софтуерна архитектура на продукта, както и дизайн на базата от данни

Етапи в софтуерната разработка

Разработка

- Най-дългият етап в софтуерната разработка
- След определяне дизайна на продукта работата се разпределя на малки задачи
- Всеки разработчик получава собствени задачи
- Всеки разработчик изпълнява своите задачи

Етапи в софтуерната разработка

Тестване

- След като софтуерът е разработен, той се подлага на тестване
- Софтуерът не преминава в следващ етап на разработка, ако тестърите не го одобрят
- Намерените бъгове и проблеми се документират и софтуерът се връща в етап на разработка

Етапи в софтуерната разработка

Инсталация и поддръжка

- Софтуера се предава на клиентите, за да се инсталира на техните устройства
- След това софтуера преминава в етап на поддръжка
- Поправят се възникнали грешки
- Софтуера се подновява от време на време

Методологии за разработка на софтуер

- Waterfall
- Agile (гъвкава методология)
- Scrum
- Kanban
- Непрекъснатата интеграция
- И др.

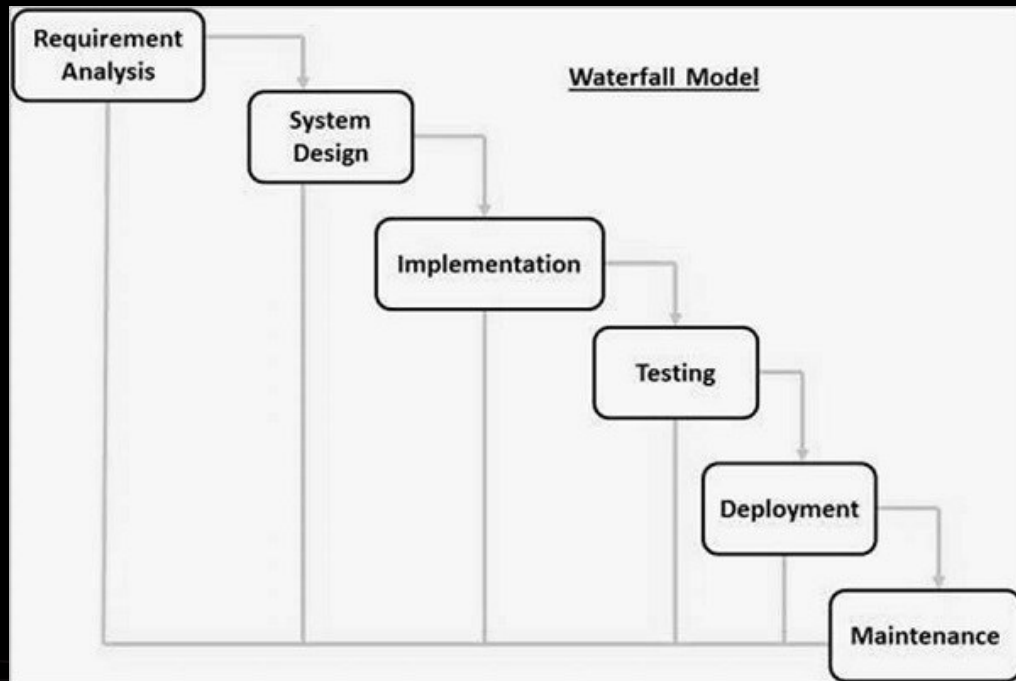
Методологии за разработка на софтуер

Waterfall

- Всеки етап трябва да приключи преди да започне следващия
- Няма припокриване между етапите
- Линеарен подход за разработка на софтуер
- По принцип резултата на даден етап играе ролята на начало за следващия
- Състои се от вече споменатите етапи на софтуерната разработка

Методологии за разработка на софтуер

Waterfall



Методологии за разработка на софтуер

Waterfall

- Всеки софтуер е различен и изисква подходящ метод за разработката си
- Waterfall моделът е най-подходящ в случаите, в които:
 - Изискванията са много добре документирани, ясни и фиксирани
 - Дефиницията на продукта е не променяща се
 - Технологиата е изяснена и не се променя динамично
 - Няма двусмислени изисквания
 - Проектът е кратък

Методологии за разработка на софтуер

Waterfall - предимства

- Лесна за разбиране и използване методология
- Лесна за менажиране поради устойчивостта на модела
- Етапите се обработват един по един
- Ясно дефинирани етапи
- Лесни за раздаване задачи
- Процеса и резултатите са добре документирани

Методологии за разработка на софтуер

Waterfall - недостатъци

- Няма работещ софтуер до самия край на последния етап
- Голяма доза риск и несигурност
- Не е добър модел за комплексни и обектно-ориентирани проекти
- Лош модел за дълги и продължителни проекти
- Не е подходящ за проекти, при които условията ще се изменят в бъдеще
- Трудно е да се измери прогрес в контекста на етапите

Методологии за разработка на софтуер

Agile

- Комбинация от итеративни и инкрементални модели за разработка
- С фокус върху задоволяване желанията на клиента чрез бърза доставка на работещи сегменти от продукта
- Разбива продукта на малки инкрементални части
- Тези части се снабдяват в итерации
 - Всяка итерация трае средно между 1 и 3 седмици
 - Всяка итерация включва вече споменатите етапи на разработка на софтуер
 - След всяка итерация работещият продукт се показва на клиента и заинтересованите страни

Методологии за разработка на софтуер

Agile

- Умение да се реагира на промени
- Основно се характеризира с адаптивност

Методологии за разработка на софтуер

Agile - манифест

- **Хората и комуникацията** стоят над процесите и инструментите
 - Самоорганизацията и мотивацията имат важна роля в гъвкавата методология
- **Работещият софтуер** стои над подробната документация
 - Работещ софтуер е по-полезен по време на срещи с клиента отколкото документацията
- **Сътрудничеството с клиента** стои над преговорите по договора
 - Непрекъснатото участие на всички заинтересовани страни е ключово
- **Адаптацията** стои над следването на определен план

Методологии за разработка на софтуер

Agile - манифест

- Манифеста на гъвкавата методология се основава на 12 принципа
- 1. Удовлетворяване на клиентите чрез бърза доставка на важен софтуер
- 2. Промяна на спецификациите дори в късните етапи на проекта
- 3. Предоставяне на работещ софтуер на периоди от по 1 до 3 седмици
- 4. Основна мярка за напредък е работещият софтуер
- 5. Ежедневно сътрудничество между бизнес служители и разработчици
- 6. Устойчива разработка, при която се поддържа постоянно темпо на работа

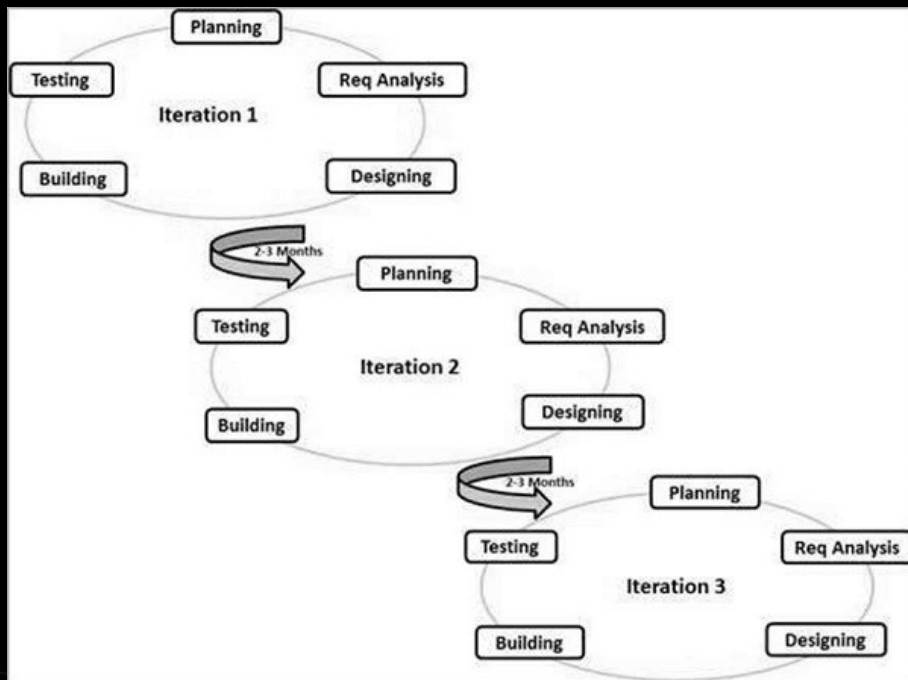
Методологии за разработка на софтуер

Agile - манифест

7. Предпочитат се разговорите лице в лице
8. Проекта се изгражда с мотивирани хора, на които се има доверие
9. Постоянно внимание насочено към добър дизайн и техническо съвършенство
10. Стремех към простота - максимално избягване от вършене на допълнителна работа
11. Екипи, които се организират сами
12. На определени интервали от време екипите се самокоригират (решават как могат да бъдат по ефективни) и съответно се адаптират към промените

Методологии за разработка на софтуер

Agile



Методологии за разработка на софтуер

Agile - предимства

- Промотира екипна работа
- Функционалността може да бъде възпроизведена бързо и съответно демонстрирана след кратък срок
- Минимални ресурсни изисквания
- Подходяща методология както за непроменящи се, така и за често променящи се изисквания
- Позволява конкурентна разработка
- Позволява работата да започне с малко или с почти никакво предварително планиране
- Лесно се управлява

Методологии за разработка на софтуер

Agile - предимства

- Дава гъвкавост на разработчиците

Методологии за разработка на софтуер

Agile - недостатъци

- Разчита на интеракция с клиент
 - Ако клиента не е наясно с изискванията си, екипът може да бъде подведен в грешна посока
- Изключително се разчита на индивидуални личности от екипа
 - Защото липсва или има минимална документация
- Може да е много трудно за нови разработчици да влязат в екипа
 - Отново поради липса на подробна документация

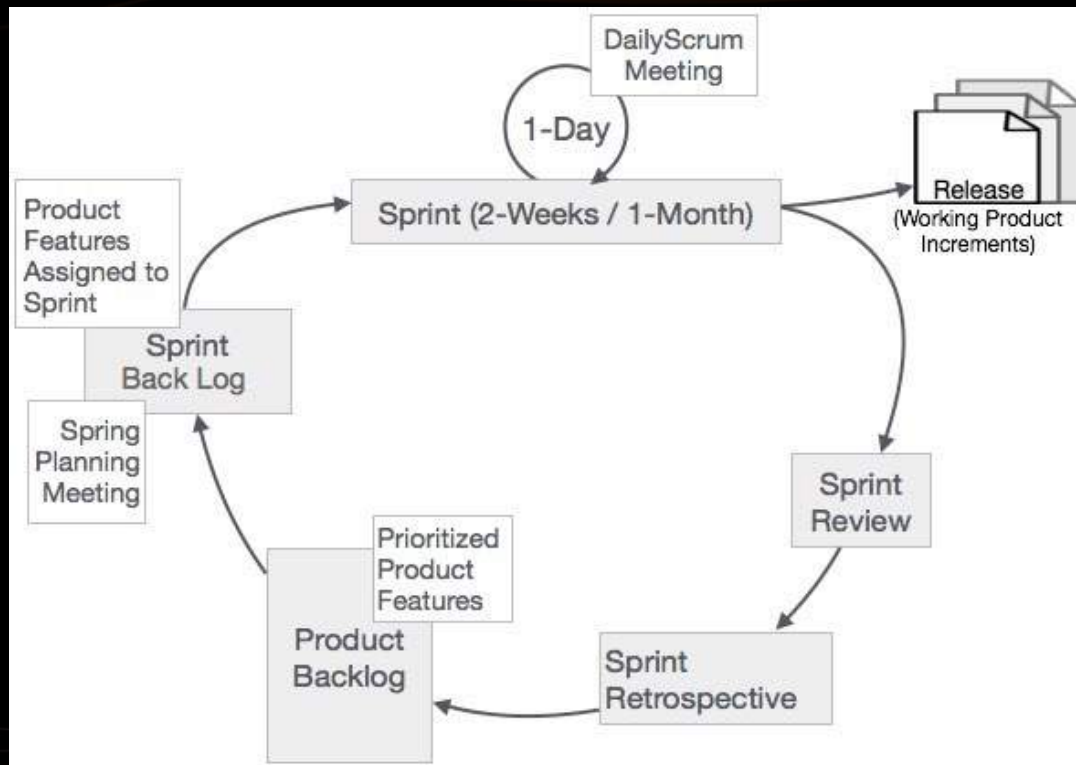
Методологии за разработка на софтуер

Scrum

- Най-популярният agile фреймуърк
 - Фокусиран върху това как да се управляват задачи в среда за разработка базирана върху екипна работа
- Използва итеративен и инкрементален подход за разработка
- Кратки периоди на итерациите
- Сравнително лесна имплементация
- Фокус върху бързите и чести доставки на работещ софтуер
- Състои се от Scrum екипи и техните роли, събития, артефакти и правила
 - Всеки компонент във фреймуърка има специфично значение

Методологии за разработка на софтуер

Scrum



Методологии за разработка на софтуер

Scrum - спринт

- Сърцето на Scrum са спринтовете
 - Времеви период от между 2 до 4 седмици, през които се разработва продукт за потенциално пускане в работа (release)
 - Нов спринт започва веднага след края на предишния
 - Спринта се състои от спринт планиране, дневни scrum срещи, работата по разработката на софтуер, ревю на спринта и спринт ретроспекция

Методологии за разработка на софтуер

Scrum - спринт

- По време на спринт планирането се планира работата, която трябва да се свърши, като това се случва в сътрудничество със scrum екипа
- Scrum срещата е ежедневно 15-минутно събитие, в което scrum екипа синхронизира работата си и изгражда план за деня
- Спринт ревюто се състои накрая на спринта, за да се инспектира новосъздадената функционалност и при нужда да се променят изисквания по продукта
- Спринт ретроспекцията се състои веднага след спринт ревюто и преди започването на следващия спринт
 - Scrum екипа анализира работата си и изгражда план за подобрене на ефективността си

Методологии за разработка на софтуер

Scrum - роли

- ScrumMaster - отговорен е
 - Процесът да върви гладко
 - Да премахва пречките, които намаляват продуктивността
 - Да организира срещите от ключова важност
- Собственик на продукта - отговорен е
 - Изискванията да бъдат разбираеми за всички
 - Да подреди в подходящ ред задачите и изискванията, за да се постигне най-добър резултат
 - Да подсигури видимостта и яснотата на изискванията, като показва на екипа по какво ще работи в бъдеще

Методологии за разработка на софтуер

Scrum - роли

- Подсигурява, че екипът разбира условията на достатъчно добро ниво
- Екип
 - Екипът се самоорганизира
 - Хора от различни сфери на дадена организация работят заедно в екипа
 - Достатъчно малък, за да остане гъвкав и подвижен
 - Достатъчно голям, за да свърши достатъчно работа по време на спринт (6 - 9 души)

Методологии за разработка на софтуер

Kanban

- Визуален начин за управление на задачи и работен процес
- Използва kanban табло с колони и карти
 - Всяка карта представлява определена задача
 - Колоните организират задачите според техния прогрес или етапа на разработка, в който се намират
- Разработен от Toyota през 40-те
- Лесно е да се види по какви задачи работи всеки един човек в организацията
- Мениджърите на екипи могат бързо да назначат задача на даден екип, когато той е без такава

Методологии за разработка на софтуер

Kanban

- Съвместим с agile
- Използван заедно със scrum, образува мощен подход за разработка на софтуер

Методологии за разработка на софтуер

Kanban - карти

- Живеят на kanban таблото
- Всяка карта представлява отделна задача
- Всяка карта е попълнена с информация относно задачата
- Всяка задача е назначена на един или няколко члена от екипа

Методологии за разработка на софтуер

Kanban - практики

- Визуализиране на работния процес
- Задаване на максимален брой задачи, по които може да се работи едновременно, с цел бързото им изпълнение
- Управление и подобряване на работния процес чрез наблюдението му и разрешаването на възникнали спънки
- Изрично уточняване на политиките за изпълнение на процеси
- Съвместна работа и експериментация - стремеж към подобрение
- Цикли за обратна връзка - ревюта, за да се достави продукта на клиента възможно най-бързо

Методологии за разработка на софтуер

Непрекъснатата интеграция

- Практика при разработката на софтуер
- Всеки екип трябва подsigурява, че успешен build и последващо тестване са проведени за всяка промяна по кода в софтуерна програма
- Концепцията гарантира премахването на проблема с откриване на грешки в по-късен етап от разработката
- Гарантира, че промените по кода никога не се правят в изолация

Методологии за разработка на софтуер

Непрекъснатата интеграция

- Отговаря на въпросите
 - Работят ли всички компоненти на софтуера заедно, както би трябвало?”
 - Минават ли всички тестове след последните промени?
 - Каква част от кода подлежи на автоматизирано тестване?

Методологии за разработка на софтуер

Непрекъснатата интеграция - как работи процеса по непрекъснатата интеграция?

- Разработчик изпраща направените промени към репозитори на система за управление на версирте. Междувременно сървърът за непрекъснатата интеграция следи репозиторито за промени (на всеки няколко минути)
- Скоро, след като промените се появят в репозиторито, сървърът за непрекъснатата интеграция ги засича и съответно извлича последното копие на кода от репозиторито, след което изпълнява build скрипт, който интегрира софтуера

Методологии за разработка на софтуер

Непрекъснатата интеграция - как работи процеса по непрекъснатата интеграция?

- Сървърът за непрекъснатата интеграция генерира обратна връзка, като изпраща резултатите от build-a по имейл на определени членове от екипа
- Пускат се unit тестове върху проекта, като се следи дали те преминават успешно
 - Ако тестовете са успешни, кодът е готов да бъде пуснат в употреба
- Сървърът за непрекъснатата интеграция продължава да следи за промени в репозиторията на системата за управление на версии и процеса се повтаря

Инструменти

- Jira

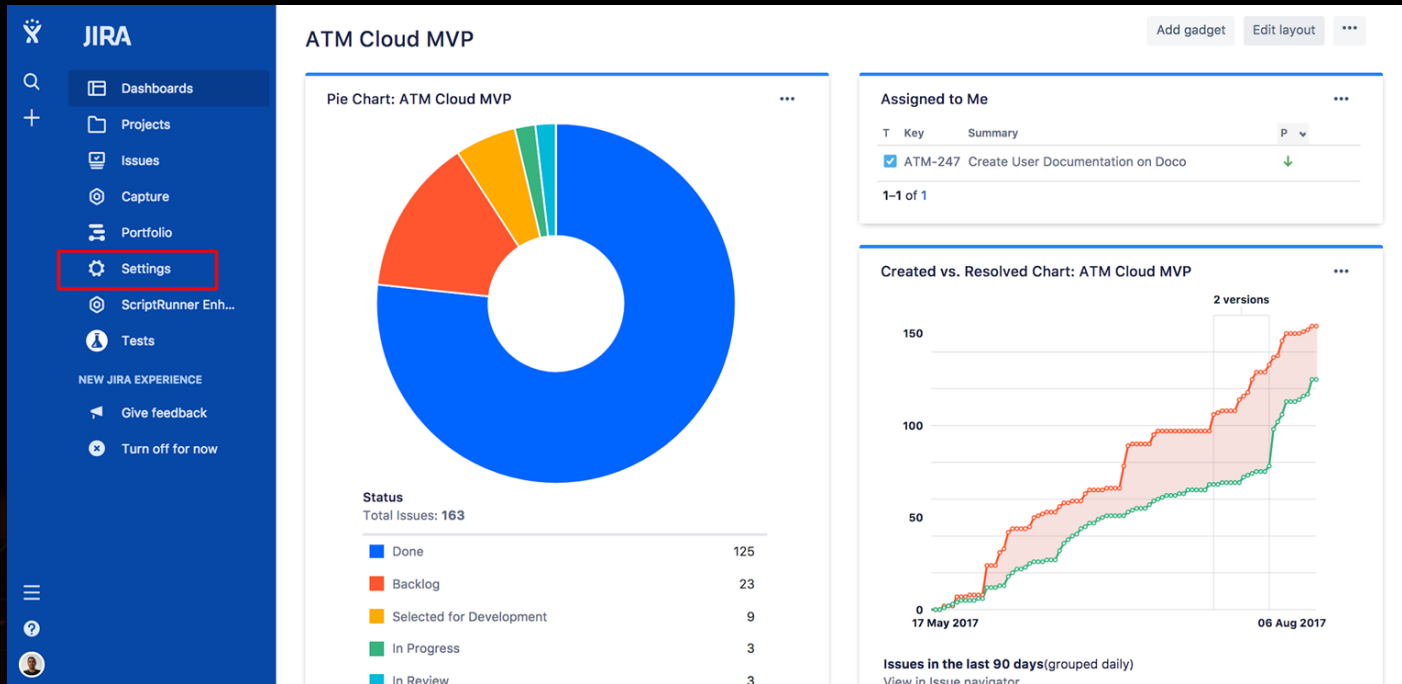


- Trello



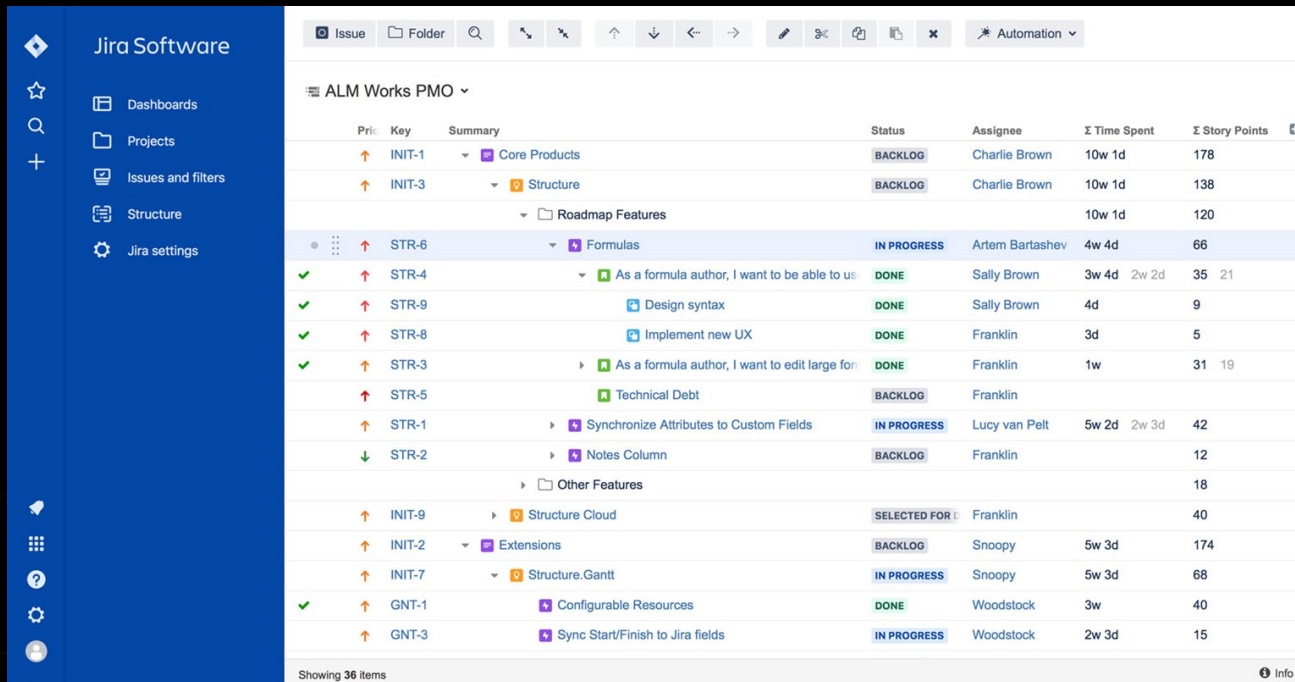
Инструменти

Jira



Инструменти

Jira



The screenshot displays the Jira Software interface. On the left is a blue sidebar with navigation icons and labels: 'Jira Software', 'Dashboards', 'Projects', 'Issues and filters', 'Structure', and 'Jira settings'. The main content area shows a list of issues under the 'ALM Works PMO' project. The issues are organized into a table with columns for Priority, Key, Summary, Status, Assignee, Time Spent, and Story Points. The issues are grouped by hierarchy, including 'Core Products', 'Structure', 'Roadmap Features', 'Formulas', 'Technical Debt', 'Synchronize Attributes to Custom Fields', 'Notes Column', 'Other Features', 'Structure Cloud', 'Extensions', 'Structure.Gantt', 'Configurable Resources', and 'Sync Start/Finish to Jira fields'.

Priority	Key	Summary	Status	Assignee	Σ Time Spent	Σ Story Points
↑	INIT-1	Core Products	BACKLOG	Charlie Brown	10w 1d	178
↑	INIT-3	Structure	BACKLOG	Charlie Brown	10w 1d	138
		Roadmap Features			10w 1d	120
↑	STR-6	Formulas	IN PROGRESS	Artem Bartashev	4w 4d	66
↑	STR-4	As a formula author, I want to be able to use	DONE	Sally Brown	3w 4d 2w 2d	35 21
↑	STR-9	Design syntax	DONE	Sally Brown	4d	9
↑	STR-8	Implement new UX	DONE	Franklin	3d	5
↑	STR-3	As a formula author, I want to edit large for	DONE	Franklin	1w	31 19
↑	STR-5	Technical Debt	BACKLOG	Franklin		
↑	STR-1	Synchronize Attributes to Custom Fields	IN PROGRESS	Lucy van Pelt	5w 2d 2w 3d	42
↓	STR-2	Notes Column	BACKLOG	Franklin		12
		Other Features				18
↑	INIT-9	Structure Cloud	SELECTED FOR	Franklin		40
↑	INIT-2	Extensions	BACKLOG	Snoopy	5w 3d	174
↑	INIT-7	Structure.Gantt	IN PROGRESS	Snoopy	5w 3d	68
↑	GNT-1	Configurable Resources	DONE	Woodstock	3w	40
↑	GNT-3	Sync Start/Finish to Jira fields	IN PROGRESS	Woodstock	2w 3d	15

Showing 36 items

Инструменти

Jira

The screenshot displays the Jira web interface for a project named 'Ardea'. The left sidebar contains navigation options: 'Issues and filters', 'My open issues', 'Reported by me', 'All issues', 'Open issues' (selected), 'Done issues', 'Viewed recently', 'Created recently', 'Resolved recently', 'Updated recently', and 'View all filters'. The main content area shows the 'Open issues' list for the 'Ardea' project. A search bar and 'Advanced search' link are at the top. Below the search bar, a list of issues is shown, with the first issue selected: 'Source string issue: This 'file' does not exist: %1\$S' (AR-2). The issue details panel on the right shows the issue title, status (To Do), assignee (Viktor Fedorovich), reporter (Kelly Wonder), and priority (High). The description of the issue is: 'Source string: This 'file' does not exist: %1\$S. Context: Description Localization. File: /random/Java.properties. Link: https://crowdin.com/translate/ardeas/28/en-uk#316'. The activity section shows a comment from Kelly Wonder: 'You have an unsaved comment'.

Projects / Ardea

Open issues

Give feedback

Back to project

Issues and filters

- My open issues
- Reported by me
- All issues
- Open issues**
- Done issues
- Viewed recently
- Created recently
- Resolved recently
- Updated recently

View all filters

Priority

Source string issue: This 'file' does not exist: %1\$S

AR-2

Source string issues reported in Crowdin

AR-1

AR-1 / AR-2

Source string issue: This 'file' does not exist: %1\$S

Attachment

Description

Kelly Wonder (kv) reported Source string issue and said: — Context doesn't correspond to the actual string translated. Please review.

Source string: This 'file' does not exist: %1\$S

Context: Description Localization

File: /random/Java.properties

Link: <https://crowdin.com/translate/ardeas/28/en-uk#316>

Activity

Comments

You have an unsaved comment

STATUS

To Do

ASSIGNEE

Viktor Fedorovich

REPORTER

Kelly Wonder

LABELS

None

PRIORITY

High

Show more

Created 31 minutes ago

Updated 34 seconds ago

Issue 1 of 2

Инструменти

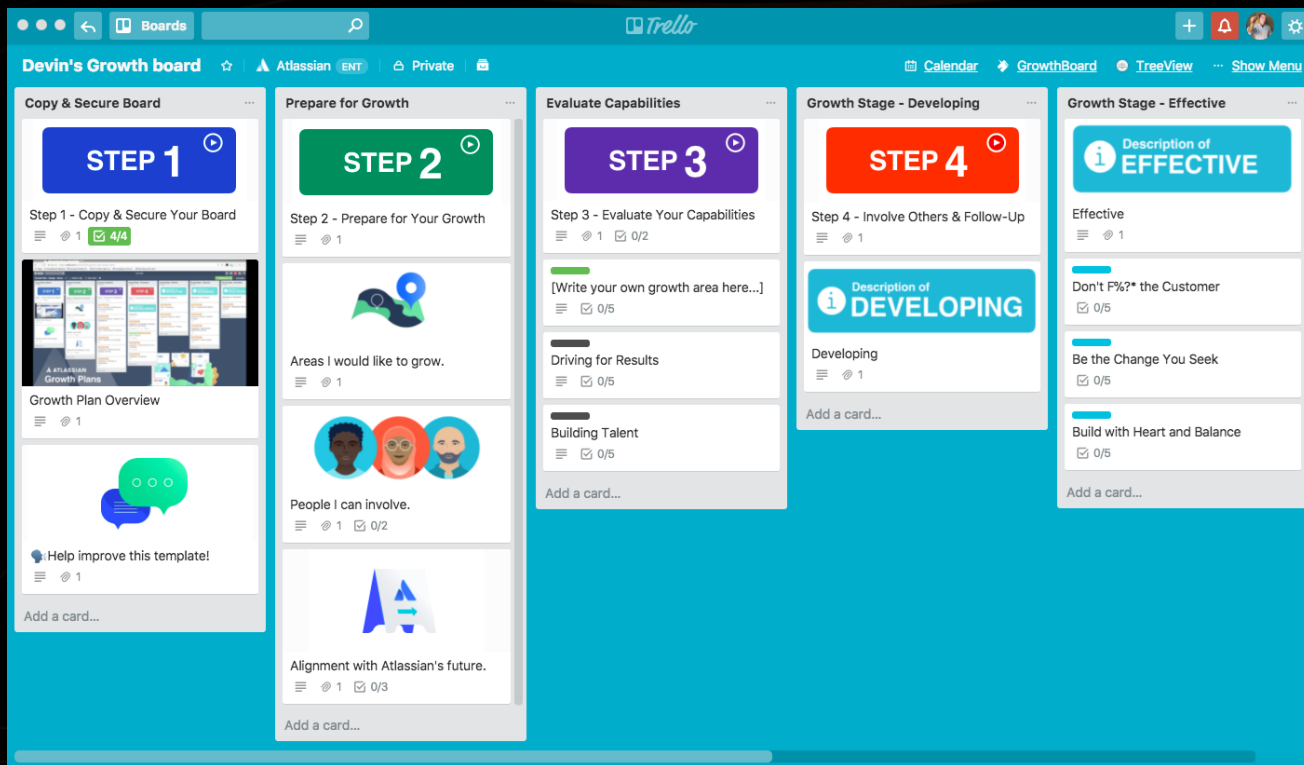
Jira

- Уеб-базирана система за проследяване на грешки, проблеми и управление на agile проекти
- Подходяща за управление и подобрене на процеси
- Предлага scrum и kanban таблици
- Известия по имейл
- Доклади за прогрес
- Ограничаване на достъпа според роля
- Следене на проблемите
- <https://www.atlassian.com/software/jira/guides/getting-started/basics?tab=classic#step-1-create-a-project> - как да започнем с

Jira

Инструменти

Trello



Инструменти

Trello

The image displays the Trello web interface. On the left, a 'Project Planning Overview' board lists various tasks. A red arrow points from a 'Website Redesign' card in this list to a larger, detailed view of the same card on the right.

Project Planning Overview

- Current Projects**
 - Update Help Documentation
 - Website Redesign** (Nov 10, 1 comment, 1 like, 27 views)
- Forecasted Projects**
 - International Sales (Dec 15)
 - Community Forum (Dec 30)
 - Equipment Update (Nov 12)
 - Partnership Opportunities (Dec 18)
 - User Research (Dec 7)
 - Analytics Data (Nov 13)

Website Redesign Card Details:

- Title:** Website Redesign
- Location:** in list [Current Projects](#)
- Members:** [Avatar 1], [Avatar 2] +
- Labels:** Marketing
- Due Date:** Nov 10 at 12:00 PM
- Description:** [Edit](#)
Goals: Our brand & product have evolved over the past two years, and our website should be updated to reflect this. The new site will be mobile-first, responsive and lightweight.
- Attachments:**
 - [Travidux Website Redesign Mockup.jpg](#) (Added a minute ago - [Comment](#) - [Delete](#))
[Download](#) [Remove Cover](#)
 - [Add an attachment...](#)
- Checklist:**
 - ☒ **Checklist** [Hide completed items](#) [Delete...](#)
 - 29%
 - ☒ Wireframe
 - ☒ Initial Design
 - ☐ Design Review
- Actions:**
 - [Add](#)
 - Members
 - Labels
 - Checklist
 - Due Date
 - Attachment
 - [Move](#)
 - [Copy](#)
 - [Subscribe](#)
 - [Archive](#)
 - [Share and more...](#)

Инструменти

Trello

- веб-базирано приложение за управление на проекти
- Използва Kanban модел
- Лесен за работа
- Безплатен софтуер
- <https://trello.com/guide/trello-101> - как да започнем с Trello

Обобщение

- Етапи в софтуерната разработка
- Методологии за разработка на софтуер
- Инструменти