

Процеси за софтуерна разработка

Съдържание

- Преговор: Етапи в софтуерната разработка
- Преговор: Waterfall методология
- Преговор: Agile методология
- Преговор с допълнение: Scrum
- Преговор с допълнение: Kanban
- Инструменти за управление на проекти и задачи

Преговор: Етапи в софтуерната разработка

- С какво се характеризира всеки етап?
- Какво включва в себе си?

Преговор: Етапи в софтуерната разработка

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

Преговор: Waterfall методология

- Какво е Waterfall?
- От какви етапи се състои?
 - Припокриват ли се те?
- Кога е подходящо да се използва тази методология?
- Какви са предимствата на Waterfall?
- Какви са недостатъците на Waterfall?

Преговор: Agile методология

- Какво е Agile?
- Къде стои фокуса, когато работим с Agile?
- С какво се характеризира?
- Какъв е Agile манифестът?
- Какви са предимствата на Agile?
- Какви са недостатъците на Agile?

Преговор с допълнение: Scrum

- Какво е Scrum?
- Как протича процеса на работа в Scrum?
- Какво е спринт?
- Какво е спринт планиране?
- Какво е Scrum среща?
- Какво е спринт ревю?
- Какво е спринт ретроспекция?

Преговор с допълнение: Scrum

- Какви са ролите в Scrum?
 - За какво е отговорен всеки един заемащ дадена роля?

Scrum

- Разгръщане пълния потенциал на Scrum [scaling Scrum]
 - В кои аспекти да се разгърне потенциала?
 - Защо бихме искали това?
- Компаниите разработват софтуер:
 - За да направят пари
 - За да защитят вече съществуващ бюджет
 - За да защитят приходи

Scrum: Scaling Scrum

- Фокусирайте се върху стойността
 - Разработвайте първо най-стойностните неща
 - Понякога това са най-рисковите неща
 - Това е отговорност на собственика, която по същество се разпределя измежду целия Scrum екип

Scrum: Scaling Scrum

- Стойността включва уменията за поддържане
 - Жертването на качество сега винаги се превръща в жертва на стойността в бъдещето
 - Екипът от разработчици трябва да бъде способен да поддържа техническо съвършенство
 - Без техническо съвършенство
 - Появява се технически дълг*
 - Стойността намалява

*технически дълг ще наричаме пречките за добавяне на нова функционалност от екипа разработчици

Scrum: Scaling Scrum

- Добавяйте нова стойност възможно най-скоро
 - Позволява по-рано да се достигне до етап, в който вече не е нужно да се работи по нова функционалност, защото клиента е задоволен
 - Тогава екипът от разработчици е свободен да работи по нещо друго с по-голяма важност
 - Същността на Agile методологията

Scrum: Scaling Scrum

- Как измерваме стойността?
 - Парите, които имаме сега са по-ценни от парите, които можем да имаме утре
- Net Present Value = мярка за всички пари, които са изхарчени или спечелени от Scrum екип (позитивна или отрицателна стойност)
 - Колкото по-късно са спечелени дадени пари, толкова по-малка стойност имат
 - Следователно колкото по-рано са спечелени, толкова по-голяма стойност ще имат

Scrum: Scaling Scrum

- Върху какво да наблегне един Scrum екип, за да разгърне потенциала си?
 - Върху нещата, които директно допринасят за по-бързото доставяне на стойност
 - Build автоматизацията и непрекъснатата интеграция са от изключителна важност
 - Само самият екип може да даде напълно правилен отговор

Scrum: Scaling Scrum

- Scrum артефакти
- Product Backlog – където се пази всичко, върху което даден екип може някой ден да работи
- Sprint Backlog – планът, който Scrum екипът създава за следващия спринт
- Increment – изхода (продукта) след всеки спринт
- Definition of Done* - дефиниция за това кога нещо се счита за свършена работа

* не е официален артефакт

Scrum: Scaling Scrum

- Scrum: процесът
- Спринт
 - Планиране на спринта
 - Всекидневна Scrum среща
 - Ревю на спринта
 - Ретроспекция на спринта
- Целта на даден спринт е константна за времевия период на спринта

Scrum: Scaling Scrum

- Екип от екипи
 - Създава се за специфичен контекст
 - Делегатите представят интересите на собствените си екипи
 - Разпуска се, когато вече няма нужда от него
 - Формира структурата на организационната мрежа
- Екип от екипи в Scrum – съществува, за да помогне на Scrum екипите в изграждането на работещи софтуерни инкременти през всеки спринт

Scrum: Scaling Scrum

- Собственик на продукта
- Управлява Backlog за продукта
 - Поддържа задачите
 - Осигурява видимостта и прозрачността на задачите
 - Подсигурява задачите да са добре разбрани от екипът разработчици
- Откриване на следващото най-стойностно нещо
 - Добрите собственици на продукт са добри мениджъри
 - Добри изследователи

Scrum: Scaling Scrum

- Екип от собствениците на продукти
 - Задължителен
 - Този екип трябва да работи, изпреварвайки срещите за планиране на спринт, за да е подготвен
- Най-важната отговорност, която собственика на продукт поема е да реши кое е следващото най-стойностно нещо
- Главен собственик на продукт – кое е следващото нещо, което да се прави, кое да не се прави, бюджет и др.

Scrum: Scaling Scrum

- Екип от разработчици
 - Допринасяне с нов инкремент след всеки спринт
 - Самоорганизация
 - Провеждане на дневна Scrum среща
 - Управление на Backlog за спринта
 - Управление на Definition of Done
- По-бързо доставяне на скорост
 - Запазване на висока техническо съвършенство
 - DevOps

Scrum: Scaling Scrum

- Всеки екип от разработчици може да забърза процеса по доставка на работещ софтуер, като използва практики по екстремно програмиране (Extreme Programming – XP) в контекста на Scrum

Scrum: Scaling Scrum

- Практики в екстремното програмиране
 - Обратна връзка
 - Test Driven Development
 - Тестване директно от клиент
 - Планувани release-и
 - Непрекъснат процес
 - Непрекъснатата интеграция
 - Подобряване на дизайна
 - Споделено споразумение
 - Стандарт в писането на код
 - Прост дизайн

Scrum: Scaling Scrum

- Екип от екипи за интеграция
 - Задължителен за големи продукти
 - Работи изпреварвайки екипите, които работят по функционалност, за да е подготвен

Scrum: Scaling Scrum

- Екип от екипи за обемно рефакториране
 - Снабдява екипа от разработчици с нови архитектури, шаблони и техники
 - Включва опитни разработчици (Senior Developers)
 - Използва Scrum, за да създаде архитектурата на системата за другите Scrum екипи

Scrum: Scaling Scrum

- Scrum Master

- Отговаря за разбирателството в екипа
- Екипите са теория, практика и правила
- Премахва пречките в един Scrum екип
- Обучава членовете на екипа
- Подпомага провеждането на Scrum събитията
- Води и учи на Scrum практики и похвати

Scrum: Scaling Scrum

- Scrum Master не е
 - Способен да гарантира успех
 - В по-висока йерархична позиция от другите членове на екипа
 - Мениджър на проект

Scrum: Scaling Scrum

- Scrum Master е отговорен за бързото добавяне на стойност от Scrum екипа
- В даден момент един Scrum Master може да бъде
 - Треньор
 - Учител
 - Ментор
 - Лидер
 - Помощник
 - Посредник

Scrum: Scaling Scrum

Scrum екип

- Основни отговорности
 - Бързото добавяне на стойност
 - Използването на Scrum
 - Учене и растеж чрез експериментиране
 - Запазването на прозрачност в работата

Scrum: Scaling Scrum

„Половината живот на доверието трае 6 седмици“
– Стив Макконъл

- На всеки 12 седмици някой трябва да се качи на самолет
 - Срещите лице в лице са за предпочитане
 - Целият екип трябва да работи заедно – лоша практика е да има членове на екипа, които работят дистанционно

Scrum: Scaling Scrum

Спринт

- Колкото по-дълго даден екип чака, преди да интегрира работата си, толкова по-голям е рискът тя да не бъде интегрирана навреме за ревюто на спринта

Scrum: Scaling Scrum

Планиране на спринт

- Преформулира се Definition of Done въз основа
 - Преглеждане на Product Backlog
 - Инкремента от последния спринт
- Споделена цел за спринте между различните екипи
- Намаляване обвързаността на екипите един с друг
- Подсигуряване на добри кадри в екипа за интеграция
- Подсигуряване, че Scrum екипите са правилно сформирани според работата, която извършват

Scrum: Scaling Scrum

Дневна Scrum среща

- Изготвяне на план за спринта, като се имат предвид
 - Целта на спринта
 - Работата, свършена вчера
 - Възможни пречки, блокери
- Целта на дневната Scrum среща е да се изготви план за следващите 24 часа
- Целта на спринта е неизменима, докато той протича

Scrum: Scaling Scrum

Ревю на спринт

- Създаване на нов product backlog, като се имат предвид
 - Заинтересованите страни
 - Scrum екипите
 - Инкремента на работещия софтуер
- В идеалния случай ревюто на спринт е просто последователност от демота, като се използва интегрирания инкремент, за да се получи обратна връзка

Scrum: Scaling Scrum

Ретроспекция на спринт

- Вземат участие всички scrum екипи, както и заинтересованите страни
- Въз основа поетите ангажменти през изминалия спринт се решава какви ще са ангажиментите през следващия, както и нов Definition of Done (в идеалния случай)
- Примери за ангажменти
 - Всяка задача от product backlog да има съответни тестове
 - Кодът да е написан според споделен стандарт за всички

Scrum: Scaling Scrum

Ретроспекция на спринт

- Scrum master-и се събират, за да споделят резултати
- Разпознаване на поведение, което може да афектира всички отбори
- Цялостно подобряване ефективността на организацията

Scrum: Scaling Scrum

Product backlog

- Просто съвкупност от задачите, които трябва да се свършат
- Управлява се от собственика на продукта
- Прозрачен е за всички
- Задачите са подредени в реда, в който трябва да се изпълнят
- Има само един product backlog

Scrum: Scaling Scrum

„Премахването на междуетипните зависимости е ключа към подобряването на продуктивността.“ – Кен Швабер

Scrum: Scaling Scrum

Инкремент

- Причината да използваме Scrum
- Интегриран, без недовършена работа, всеки спринт
- Всеки Scrum екип, който се провали в интегрирането в спринт, увеличава бъдещия риск
- Функционалност, която не е готова – не е достъпна
- Дадена функционалност може да бъде различна за различни потребители
 - Така с тестване се добива представа, кое как се възприема от клиентите

Scrum: Scaling Scrum

Definition of Done

- Използва се, за да се определи точно завършеността на работата
- Преобразува се и се разширява по време на ретроспекциите на спринт
- Може да се отнася за процес или инженерни практики
- В среда със сътрудничество, споделен Definition of Done ще възникне измежду Scrum екипите

Преговор с допълнение: Kanban

- Какво е Kanban?
- Какво позволява Kanban?
- С какво се характеризира?

Преговор с допълнение: Kanban

- Kanban или kanban?
 - Kanban и Kanban метод насочват към процеса
 - kanban насочва към определена карта, използвана в процеса

Преговор с допълнение: Kanban

- Защо да използвам Kanban?
 - Подобряване качеството на работа
 - Идентифициране и елиминиране на затрудненията
 - Намаляване времето прекарано чакане на опашка
 - Подобряване на екипната работа
 - Намаляване на напразните усилия

Преговор с допълнение: Kanban

- Оползотворяване на всички налични ресурси или забързване на процеса
 - За предпочитане е да се забърза процесът на работа пред това да се оползотворят всички от наличните ни ресурси

Преговор с допълнение: Kanban

Основни принципи

- Визуализирайте работата си
- Ограничете броя на задачите, които са в процес на работа (Work-In-Progress)

Преговор с допълнение: Kanban

Основни принципи

- Защо да визуализирате работата си
 - Позволява вие и другите да виждат с какво се занимавате
 - Намалява стреса
 - Намалява шанса да бъдат забравени важни задачи
 - Подобрява умението ви да правите добри избори
 - Върху какво трябва да работите в момента?
 - Кога да кажете „Не“ на нови задачи?
 - Кои задачи в момента са блокирани?
 - Колко време отнема, за да се свърши нова работа?

Преговор с допълнение: Kanban

Основни принципи

- Защо да ограничите броя на задачите, които са в процес на работа
 - Намалява пропиляната работа
 - Подобрява качеството
 - Спомага протичането на процеса

Преговор с допълнение: Kanban

Kanban методът

- Свойства
 - Визуализация процеса на работа
 - Ограничаване на несвършената работа (Work-In-Progress)
 - Измерване и управление на процеса на работа

Преговор с допълнение: Kanban

Kanban методът

- Рецепта за успех
 - Фокусирайте се върху качеството
 - Намалете WIP (Work-In-Progress)
 - Добавяйте към завършените задачи често
 - Приоритизирайте

Преговор с допълнение: Kanban

Kanban инструменти

- AgileZen
- Trello
- LeanKit Kanban
- targetprocess

Обобщение

- Преговор: Етапи в софтуерната разработка
- Преговор: Waterfall методология
- Преговор: Agile методология
- Преговор с допълнение: Scrum
- Преговор с допълнение: Kanban
- Инструменти за управление на проекти и задачи