**Курсов проект по дисциплината “Управление на софтуерни проекти”**

*Анализ и избор на методология за изпълнение на проекта*

Изготвил: Ерика Карамучева

Факултетен Номер: 2101321067

Специалност: Софтуерно инженерство

Съдържание:

1. Въведение
2. Анализ на методологиите

2.1. Scrum методология

2.2. Kanban

1. Сравнителен анализ спрямо ключови фактори

3.1. Времеви рамки

3.2. Екипна структура

3.3. Изисквания

3.4. Риск мениджмънт

1. Избор на методология

4.1. Основни аргументи за избора

4.2. Специфично прилагане

1. Заключение
2. Въведение

При разработката на уеб приложението Educatings, е важно да изберем методология, която ще позволи ефективното управление на проекта, както и да отговори на специфичните нужди на приложението. Ще разгледаме две методологии**- Scrum** и **Kanban**, като анализираме тяхната приложимост за нашия проект. И Scrum и Kanban принадлежат към семейството на Agile и имат за цел да подобрят ефективността на екипите при разработката на софтуер. Те споделят основни принципи за гъвкавост, бърза реакция и фокус върху сътрудничеството, но се различават по структурите и начините на работа.

1. Анализ на методологиите

2.1. Scrum методология

Scrum е итеративна, инкрементална рамка за управление на проекти, базирана на принципите на Agile. Тя предлага структуриран подход за организиране на работния процес чрез разделяне на проекта на спринтове (обикновено с продължителност между 2 и 4 седмици). В началото на всеки спринт екипът избира определени задачи, върху която да работи до неговия край. Основната цел е редовно доставяне на функционален софтуер и постоянна обратна връзка от страна на клиентите. Scrum се характеризира със строга структура и регулярност, като задачите и приоритети се преразглеждат след всеки спринт.

а) Силни страни

* **Ясна структура на работа и екипи:** Scrum може да предостави добре дефинирани етапи, в които да се работи по различни функционалности. Всеки спринт може да бъде насочен към определена част на приложението (например върху системата за управление на оценки, разработката на чат функционалностите, форумите и други).
* **Имаме работеща версия на ранен етап:** Scrum е идеален вариант за приложения, които ще предлагат много функционалности (подобно на нашето приложение). Така например в началото могат да се разработят основните функционалности на приложението като функционалността за разписание и напомняния и дигиталните класни стаи. А на по- късен етап екипът може да се фокусира върху останалите изисквания- системата за управление на оценки и други.
* **Постоянна обратна връзка и корекции:** С провеждането на редовни срещи между заинтересованите страни можем да правим своевременни корекции в дизайна и функционалността. Това е особено важно за нашето приложение, защото студентите и преподавателите имат специфични изисквания.

б) Слаби страни

* **Поява на неочаквани промени:** Изискванията към приложението могат да бъдат променяни динамично (може да бъде разгледан и като силна страна). Промените трябва да бъдат планирани и да се вписват в продуктовия списък, което от своя страна може да доведе до натиск върху екипа.
* **Трудност при управлението на дългите срокове:** Ако при разработката се натъкнем на много непредсказуеми промени или неотложни задачи, Scrum може да изпита затруднения, когато необходимите корекции изискват повече време, отколкото може да бъде осигурено в рамките на спринта.

2.2. Kanban

Канбан е метод за управление на интелектуалната дейност, който набляга на доставката точно на време, без същевременно членовете на екипа да са претоварени. Методологията се фокусира върху непрекъснатото подобряване на работния процес чрез визуализиране на задачите на специални дъски. Работата се организира в етапи и задачите преминават в различни фази, като например “В процес” и “Готово”. Основният принцип на Канбан е ограничаването на работата в процеси, което предотвратява претоварване и осигурява плавен поток на задачите. Методът е подходящ за екипи, които работят по непредсказуеми задачи и имат нужда от постоянен поток на работа без твърдо определени срокове.

а) Силни страни

* **Лесно управление на задачите:** Канбан позволява лесно управление на задачите чрез визуализация на дъската за работа и лесно следене на напредъка.
* **Гъвкавост на екипа:** Канбан не изисква строга структура на екипа и срещите, както при Scrum. Това означава, че можем да се фокусираме върху основните задачи и да не губим време по административните дейности.
* **Непрекъснат поток на работа:** Методологията ни дава възможност да работим по задачите си без времеви ограничения и спринтове. Това може да бъде от голяма полза, ако имаме множество малко задачи, които не изискват специфичен спринт, като например поправки на бъгове или малки промени в интерфейса. Канбан ни позволява плавно преминаване от една задача към друга, което е удобно за екипи, които работят по множество функционалности в един и същи период.

б) Слаби страни

* **Липса на структура:** При големи проекти като нашия, планирането на изпълнението на по- големи функционалности може да се окаже сериозно предизвикателство.
* **Блокиране на задачи:** Ако не се управлява добре, ограничаването на нови задачи може да доведе до блокиране на по- важни такива.

1. Сравнителен анализ спрямо ключови фактори

3.1. Времеви рамки

А) Scrum – строго дефинирана времева рамка в рамките на спринтове ( обикновено с продължителност от 2 до 4 седмици). ( в нашия случай 14 спринта)

Б) Kanban – няма фиксирани времеви рамки. Работата се извършва с постоянен поток, без натиск за завършване в конкретни периоди.

3.2. Екипна структура

А) Scrum – строга екипна структура с ясно дефинирани роли. Всеки член има специфична роля и отговорности.

Б) Kanban – няма фиксирани роли. Екипът може да се адаптира спрямо нуждите на проекта.

3.3. Изисквания

А) Scrum – изискванията се дефинират преди всеки спринт. Те могат да бъдат променяни в края на всеки спринт.

Б) Kanban – изискванията могат да бъдат добавяни или променяни по всяко време, тъй като няма твърди времеви рамки.

3.4. Риск мениджмънт

А) Scrum – рисковете се управляват по време на регулярните срещи, които позволяват ранното идентифициране на проблемите.

Б) Kanban – менажирането на риска е по- неформално, като се управлява чрез визуализация на процеса, което минимализира натоварването.

1. Избор на методология

След задълбочен анализ, избираме Scrum методологията за изпълнение на проекта.

4.1. Основни аргументи за избора

А) Естество на проекта

- Разработката на приложението изисква честа обратна връзка и от студентите, и от преподавателите.

- Изискванията към приложението могат да еволюират.

- Честите срещи, провеждани в Scrum осигуряват редовен напредък и възможност за адаптация към изискванията.

Б) Времева рамка

- Времето, с което разполагаме- 140 дни е идеално за разделяне на 2- седмични спринтове.

- По – добър контрол върху прогреса.

4.2. Специфично прилагане

За да компенсираме слабите страни на Scrum, ще въведем следните практики:

**А) Гъвкавост на промените по време на спринта:** в Scrum промени в задачите след започване на спринта са крайно нежелателни. За да компенсираме това, ние ще използваме Kanban табло в рамките на спринта, за да управляваме задачите. Така, ако възникне задача, тя може да бъде маркирана като „критична“ и да се добави към текущия спринт. За да избегнем и последващото натоварване на екипа ще въведем и буферно време за непредвидени задачи.

**Б) Визуализация на натоварването на екипа:** в Scrum не можем да видим доколко натоварен е всеки член от екипа в даден момент. За целта ще включим Kanban метрики (като WIP - задачи в „процес“) за проследяване на натоварването. Също така ще използваме и различни диаграми, с помощта на които да анализираме производителността.

**В) Ограничаване на задачите в даден спринт:** за да избегнем напрежението на екипа от крайните срокове и бързия темп, ще ограничим задачите в процес на изпълнение. Ще въведем „гъвкави спринтове“, където част от времето ще е резервирано за почивка или работа по странични задачи.

1. Заключение

Както Scrum, така и Kanban предлагат големи възможности за разработка на софтуерни приложения. За нашето приложение ние ще се спрем на Scrum методологията, като основна такава. За да сме максимално ефективни обаче, ще въведем и някои методи от Kanban. Чрез комбинацията на двете методологии ще създадем процес на работа, който не само е структуриран и предвидим, но и гъвкав и адаптивен. Това от своя страна ще гарантира успеха на нашето приложение.