

Agile методологии (продължение) – ресюме

### **Lean Manufacturing-** Производствен процес на Toyota:

- **Just-in-time (JIT)** – производство само на това, което е необходимо, и точно когато е необходимо: Отговаря на текущите нужди, без прогнозиране за бъдещето; Без наличност на склад.
- **Системи на изтегляне (Pull systems)** –Нуждите на клиентите „изтеглят“ работата, вместо тя да бъде „натисната“ по предварително изготвен график.
- **Ранно идентифициране на дефекти** –Линията спира, ако бъде открит дефект.
- **Анализ на първопричината** – изследване и отстраняване на корена на проблема.
- **Седем вида загуби:**
  - Свръхпроизводство
  - Изчакване
  - Транспорт
  - Прекалена обработка
  - Прекомерно движение
  - Наличности (складови запаси)
  - Дефекти

**Lean Software Development-**Произлиза от **Lean-Manufacturing**.Основополагащо произведение: **Lean Software Development: An Agile Toolkit**, автори Мери и Том Попендийк (2003).

- Представлява инструментариум за преобразуване на широко приети Lean принципи в ефективни Agile практики.
- „Основата на тази книга са 22 мисловни инструмента, които подпомагат ръководителите на софтуерни разработки да разработят Agile практики, които са най-ефективни в техния конкретен контекст.“

### **Принципи и инструменти**

#### **1. Елиминиране на загубите**

- **Инструмент 1: Разпознаване на загубите** – идентифициране и премахване на ненужни дейности или ресурси, които не добавят стойност към продукта.
- **Инструмент 2: Картиране на потока на стойността (Value Stream Mapping)**
  - **Картиране на потока на стойността в Agile** – използване на Agile подходи за анализ на стойността, която всяка стъпка от процеса добавя към продукта.
  - **Традиционно картиране на потока на стойността** – идентифициране на всички стъпки в производствения или разработващ процес, за да се открият и премахнат загубите.

## 2. Усилване на ученето (Amplify Learning)

- **Инструмент 3: Обратна връзка (Feedback)** от потребителите и екипа.
- **Инструмент 4: Итерации (Iterations)** – разработване на продукта на малки, управляеми цикли.
- **Инструмент 5: Синхронизация (Synchronization)** – координиране на дейностите между различни екипи и процеси.
- **Инструмент 6: Базирано на множества развитие (Set-Based Development)** – разработване на множество опции, комуникиране на ограниченията и позволяване на решенията да се появят естествено.

## 3. Решения в последния момент (Decide as Late as Possible)

- **Инструмент 7: Мислене за опции (Options Thinking)** – запазване на повече алтернативи възможно най-дълго.
- **Инструмент 8: Последният отговорен момент (The Last Responsible Moment)** – вземане на решение в последния момент, когато пропускането му би премахнало важна алтернатива.
- **Инструмент 9: Вземане на решения (Making Decisions)** – внимателно обмисляне на решенията въз основа на наличната информация и контекста.
- **Доставка с висока скорост (Deliver Fast)**

- **Инструмент 10: Системи на изтегляне (Pull Systems)** – клиентските нужди определят кога и какво да бъде доставено.
- **Инструмент 11: Опашки (Queueing)** – управление на задачите и минимизиране на времето за изчакване.
- **Инструмент 12: Цена на забавянето (Cost of Delay)** – оценяване на въздействието от забавено доставяне на продукта.

## **5. Упълномощаване на екипа (Empower the Team)**

- **Инструмент 13: Самоопределение (Self-determination)** – даване на екипа свободата да взема решения.
- **Инструмент 14: Мотивация (Motivation)** – поддържане на ангажираността и вдъхновението на екипа.
- **Инструмент 15: Лидерство (Leadership)** – изграждане на визия и посока.
- **Инструмент 16: Експертиза (Expertise)** – развиване на умения и знания в екипа.

## **6. Изграждане на цялостност (Build Integrity In)**

- **Инструмент 17: Възприета цялостност (Perceived Integrity)** – продуктът отговаря на нуждите на потребителите и оставя добро впечатление.
- **Инструмент 18: Концептуална цялостност (Conceptual Integrity)** – продуктът е последователен и добре структуриран.
- **Инструмент 19: Рефакторинг (Refactoring)** – подобряване на кода без промяна на функционалността.
- **Инструмент 20: Тестване (Testing)** – гарантиране на качеството чрез непрекъснато тестване.
- **Инструмент 21: Измервания (Measurements)** – оценяване на ефективността и напредъка.
- **Инструмент 22: Договори (Contracts)** – създаване на ясни и ефективни споразумения между страните.

**Lean startup** - Методология за бързо установяване дали предложен бизнес модел или продукт е жизнеспособен и, ако е така, неговото итеративно подобрене в кратки цикли на разработка.

Началната точка е убеждението, че **всеки нов продукт е експеримент**, който трябва да отговори на въпроса: „**Трябва ли този продукт да бъде създаден?**“, а не „Може ли да бъде създаден?“

Основни концепции и принципи:

- **Потвърдено учене (Validated Learning)**: систематично придобиване на знания за продукта чрез реални потребителски данни.
- **Цикли „Изграждане-Измерване-Учение“ (Build-Measure-Learn)**: бързи итерации за разработка, тестване и анализ.
- **Минимално жизнеспособен продукт (MVP)**: версия на продукта с минимални функционалности, достатъчна за тестване на основната идея.
- **Действащи метрики (Actionable Metrics)**: измервания, които предоставят реална стойност и помагат при вземане на решения.
- **Пивоти (Pivots)**: промени в посоката на развитие, когато първоначалната идея не дава очакваните резултати.

**Kanban** – механизъм за планиране в системи на изтегляне (pull systems):

- Подобрена видимост на проекта.
- Насърчаване на комуникацията и сътрудничеството.
- Максимизиране на ефективността.
- Подобрена предвидимост на процесите.

Методът Kanban за софтуерна разработка предлага:

- Рамка за процеси (process framework).
- Не изисква конкретно дефинирани роли или артефакти.

Kanban – Основни практики:

- Визуализирайте работния процес (Visualize the workflow): използване на табла и карти за представяне на текущите задачи.

- Ограничете текущата работа (Limit work-in-progress): задаване на максимален брой задачи, които могат да се изпълняват едновременно.
- Измервайте и управлявайте потока (Measure and manage flow): анализиране на времето за изпълнение и идентифициране на пречки.
- Направете процесните политики ясни (Make process policies explicit): дефиниране и споделяне на правилата за работа.
- Подобряване чрез сътрудничество (Improve collaboratively): активно включване на екипа в идентифицирането и внедряването на подобрения.
- Внедряване на цикли за обратна връзка (Implement feedback loops): редовно получаване и анализиране на обратна информация за процесите.

**Scrumban** - Прилагане на практиките на Kanban в Scrum.

Характеристики:

- Включва някои от **ролите, срещите и артефактите на Scrum**.
- **Итерациите с фиксирано време** са по избор.
- Планирането, прегледите и ретроспективните срещи се провеждат при необходимост.
- Използва табло за задачи (task board) за визуализиране на работния процес.
- Не използва беклог за спринтове (sprint backlogs) или диаграми за изгаряне (burndown charts).

**ScrumXP Hybrid** - Добавяне на техническите практики на Extreme Programming (XP) към управленската рамка на Scrum.

**State of Agile Report: Разширяване на Agile (Scaling Agile)**

Първоначално разширяването на Agile се осъществява чрез подхода „**Scrum of Scrums**“ - През последните пет проучвания, употребата на **SAFe® (Scaled Agile Framework)** се е увеличила значително и сега е доминиращ подход, използван от повече от една трета от анкетираните.

**Scrum of Scrums** - Прост механизъм за мащабиране на Scrum. Всеки екип номинира представител (амбасадор), който да участва в срещи „Scrum of Scrums“ ежедневно или с по-ниска честота. Амбасадорите споделят:

- Напредъка на екипа.
- Следващите стъпки.
- Пречките, пред които е изправен екипът.
- Развиятия, които могат да предизвикат затруднения за др. екипи.

**SAFe: Scaled Agile Framework:** Прилага принципите на **Lean** и **Agile** на всички нива в една организация, включително доставката на решения, разработката на продукти и управлението на портфолио. SAFe е конфигурируем:

- **Essential SAFe** за малки решения с 50-125 специалисти.
- **Portfolio SAFe** за големи решения за комплексни системи, които изискват хиляди хора.
- **Large Solution SAFe** за решения със среден мащаб.

**Essential SAFe:** Позиционира **основните принципи на SAFe в малки и средни мащаби**, като позволява прилагането на Lean и Agile в по-малки екипи и проекти.

SAFe практики:

**Agile Teams:**

- Крос-функционални групи от 5 до 11 души, които дефинират, изграждат, тестват и доставят инкременти на стойност в кратки времеви рамки (итерация от 2 седмици).
- Приложение на практики от Scrum, XP или Kanban.
- Включва роли на Scrum Master и Product Owner.

**Agile Release Train (ART):**

- Дълготраен виртуален екип от екипи, обикновено между 50 и 125 души, които планират, ангажират се, разработват и внедряват заедно.
- ART е организиран около значими стойностни потоци на предприятието.

**Program Increment (PI):**

- Времево ограничен интервал за планиране, в който Agile Release Train планира и доставя инкрементална стойност под формата на работещ и тестван софтуер или системи.
- Обикновено 8-12 седмици (4-6 итерации).

**Innovation and Planning Iteration:** Процесът, който се провежда в края на всяко Program Increment (PI). Това е буфер за оценка, който позволява изпълнението на PI целите и осигурява време за иновации, образование, PI планиране и събития за инспекция и адаптация.

**Program Increment (PI) planning:** Събиране на всички членове на ART в рамките на 2 дни за установяване на alignment (съгласуваност) и колективни цели за следващия PI период.

**System demo:** В края на всяка 2-седмична итерация се демонстрира интегрираната работа на всички екипи от ART на заинтересованите страни, които дават обратна връзка за коригиране на посоката на работата.

**Retrospective workshop:** Провеждане на ретроспективен семинар в края на Program Increment (PI), за да се идентифицират основните пречки чрез анализ на коренните причини.