

wp

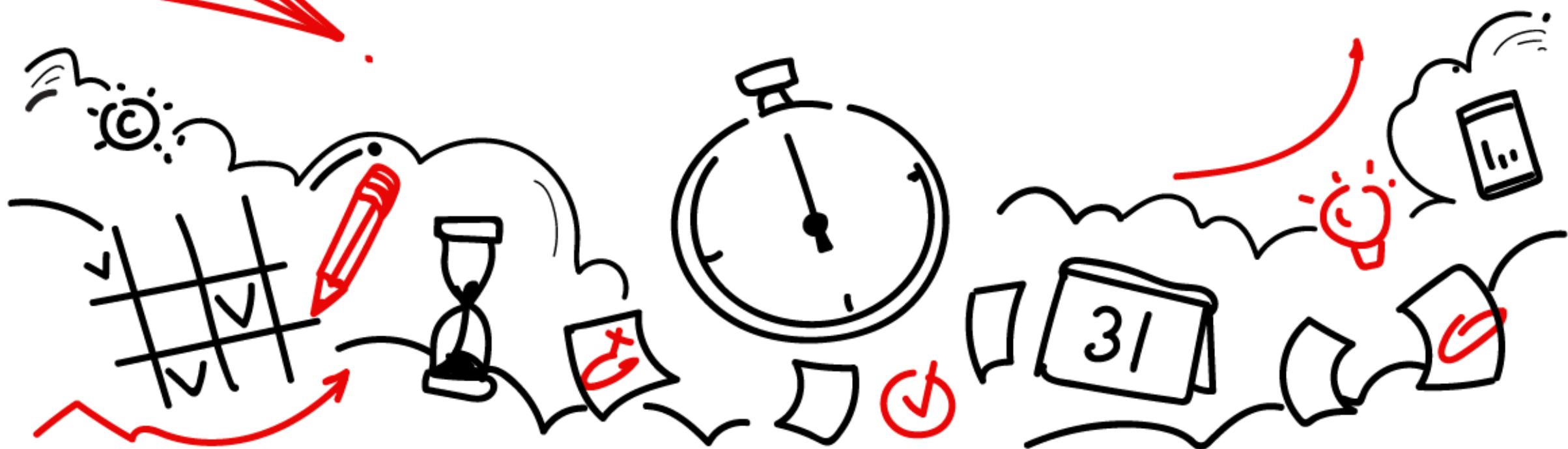
PJAKT – kurs GO





## ( Czym jest Scrum? )

- Łatwy do zrozumienia,  
trudny do opanowania -



# Zwinne metodyki zarządzania projektami



## Co to jest Agile?

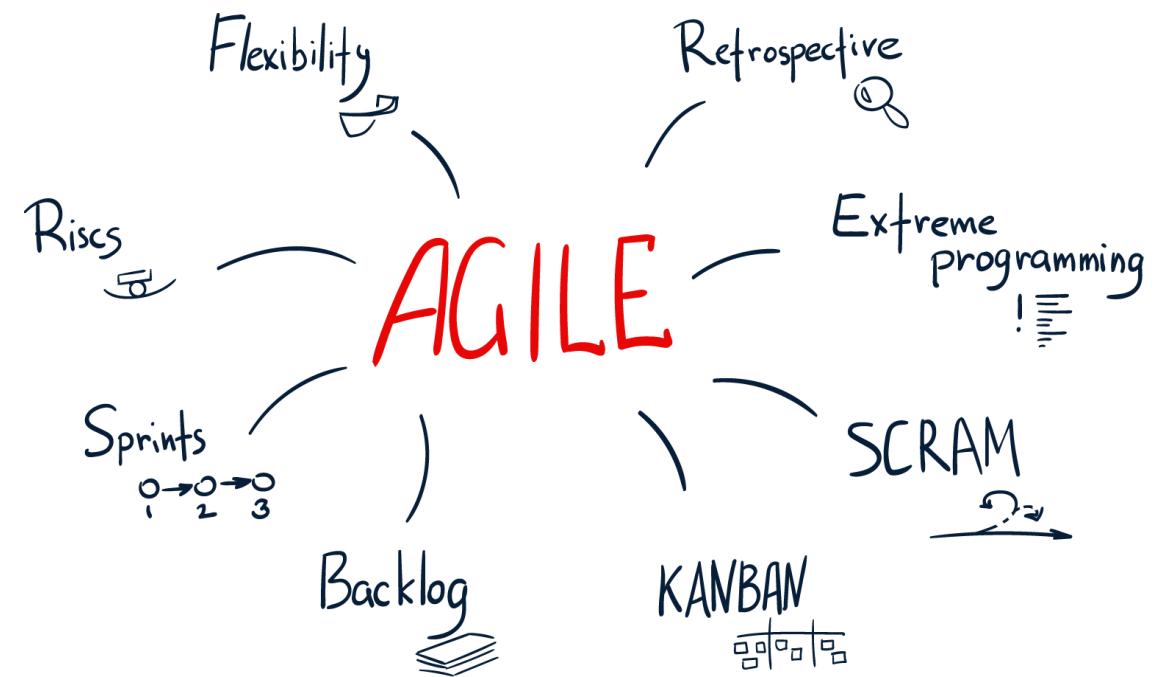
- Metodyka zarządzania projektami bazująca na podejściu iteracyjnym, mająca na celu maksymalizację wartości produktu i szybkości dostarczania go.

## Manifest Agile

- Powstała w 2001 lista postulatów mająca na celu usprawnienie procesu wytwarzania oprogramowania.

## Metody zwinne

- Scrum, Kanban, Extreme Programming, Lean,



# Scrum



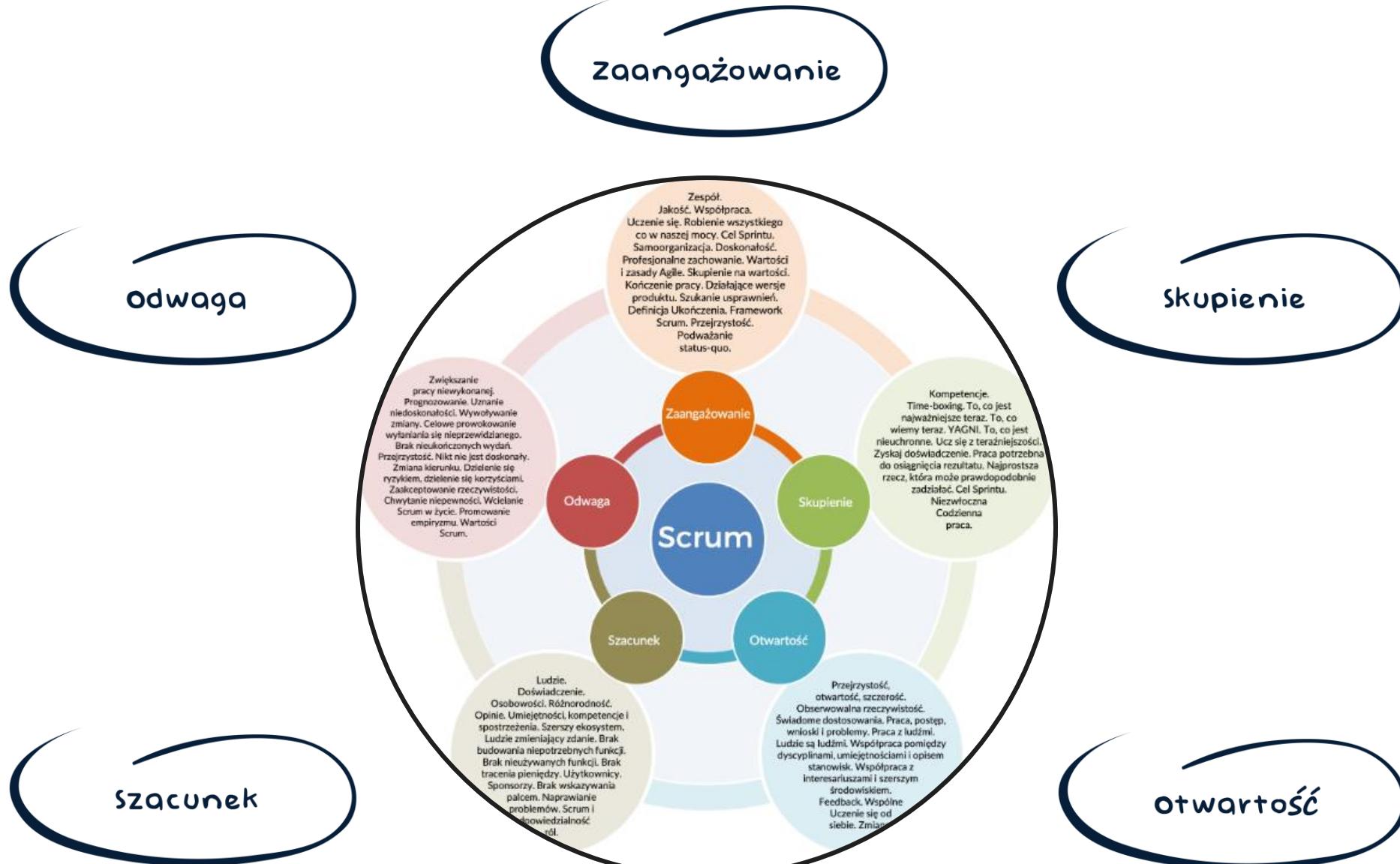
- Scrum to zwinne (Agile) podejście do **tworzenia nowych produktów i usług.**

Produktem może być dowolna rzecz.

- **Nie jest metodyką** zarządzania projektami
- **Framework** -ramy postępowania, w ramach których możemy budować procesy.
- Zgodny z **Manifestem Agile**
- Składa się z **ról, wartości, zdarzeń i artefaktów.**
- Scrum opiera się na **dostarczaniu produktu** w małych przyrostach (increment)



# Wartości Scrum



# Odpowiedzialności



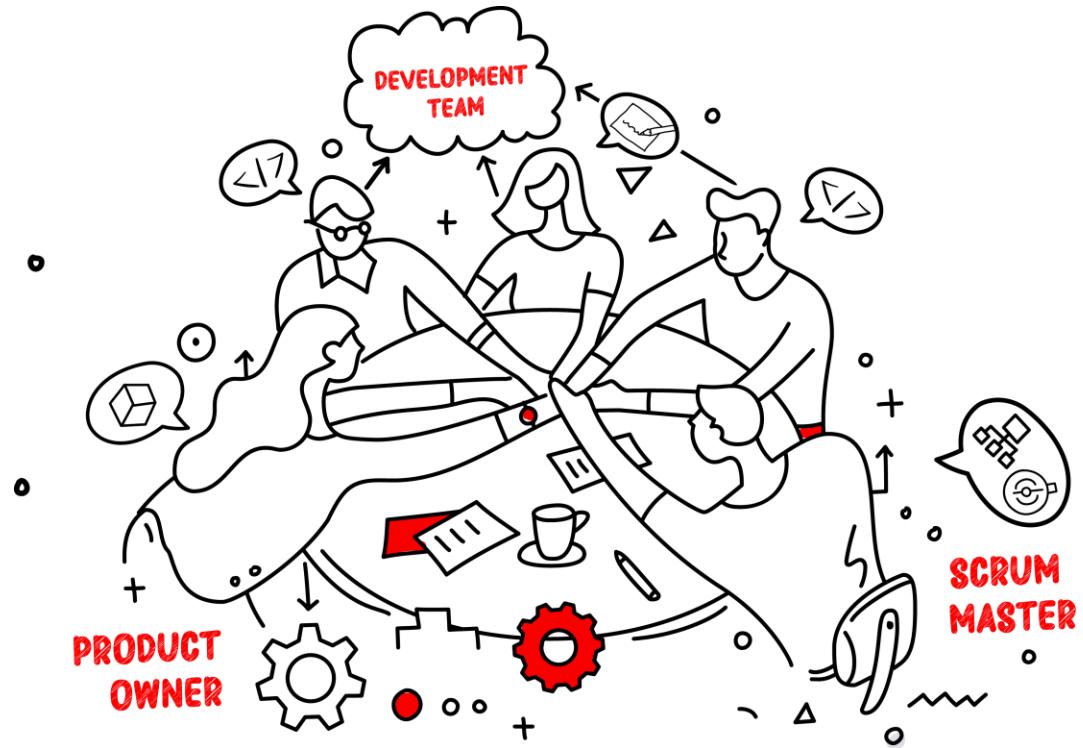
# Zespół Scrumowy

Scrum bazuje na niewielkim zespole nazywanym **Zespołem Scrumowym (Scrum Team)**.

W ramach tego zespołu wyróżniamy trzy odpowiedzialności, czyli Deweloperów (Developers), Scrum Mastera i Product Ownera.

## Najważniejsze Informacje

- Składa się Deweloperów, Product Owner, Scrum Master
- Interdyscyplinarny – mamy wszystkie kompetencje do wykonania pracy
- Samozarządzający - sam organizuje pracę
- Do 10 osób
- Brak hierarchii wewnętrz z zespołu



# PRODUCT OWNER

WP



**Właściciel Produktu** (czyli Product Owner, PO), to osoba, która ma wizję produktu i dzięki temu może decydować o tym, co jest w danym momencie najważniejsze.

## Zadania PO

- Odpowiada za backlog produktu czyli listę pomysłów, które potencjalnie mogą zostać zrealizowane w ramach prac rozwoju produktu.
- Zarządza backlogiem i jego priorytetyzacją zgodnie z wartością biznesową. Weryfikuje czy aktualnie rozwijane featury przyniosą największą korzyść biznesową.
- W trakcie sprintu pozostaje w kontakcie z zespołem deweloperskim.
- Odpowiada za wizję i cele produktu
- Może zakończyć sprint przed czasem.
- Prowadzi komunikację z interesariuszami.

# SCRUM MASTER

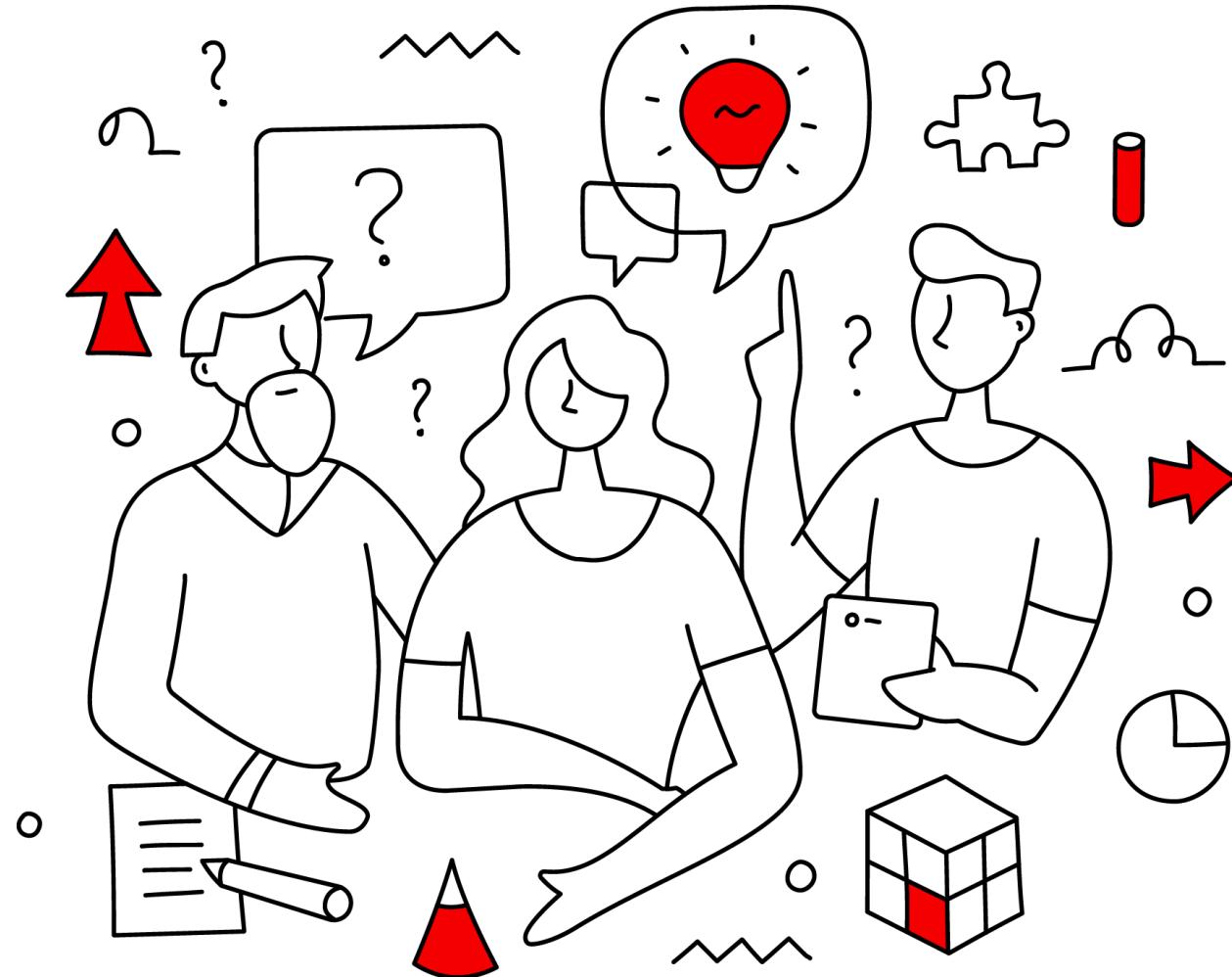


**Scrum Master** to osoba, która dba aby przebieg prac był zgodny z zasadami Scruma i ustalonymi przez zespół

## Zadania Scrum Master

- Skupia się na tym **JAK** działamy. Dba o prawidłowe wykorzystanie framework'u Scrum. Czuwa nad tym, aby przebieg prac był zgodny z zasadami Scruma i ustalonymi przez zespół
- Usuwa przeszkody, na które napotyka zespół w trakcie sprintu.
- Zarządza komunikacją i ją facylituje.
- Uczestniczy i facylituje wydarzenia scrumowe
- Servant leader

# DEWELOPERZY



**Deweloperzy** to grupa specjalistów, którzy wspólnie pracują nad stworzeniem produktu.

## Zadanie Deweloperów

- Wytworzenie wartościowego i wdrażalnego przyrostu.
- Decydują, które elementy PBI są możliwe do zrealizowania oraz jak je zrealizować.
- Mają decyzyjność w sprawie tego jak powinny być zrealizowane zadania
- Jest to zespół interdyscyplinarny, składający się z osób, które będą się wspierać w ramach prac nad celem sprintu.
- Są odpowiedzialni za backlog sprintu oraz monitorowanie postępów w realizacji celu sprintu.

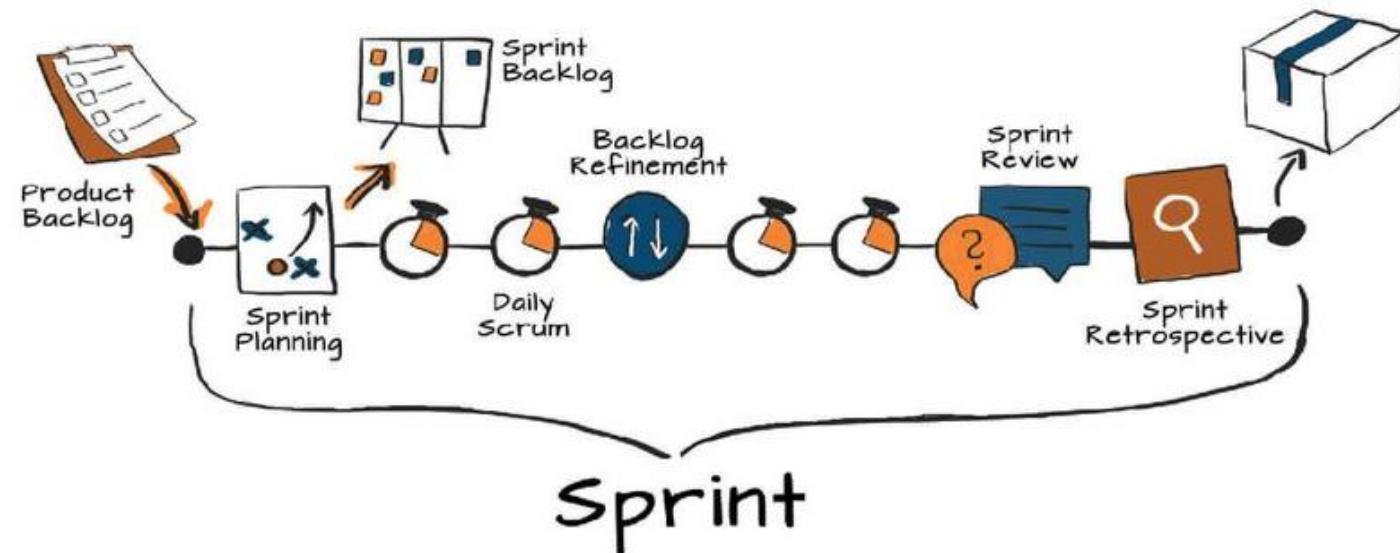
# Wydarzenia



W Scrumie odbywają się **cztery formalne wydarzenia umożliwiające inspekcję i adaptację w ramach obejmującego je wszystkie wydarzenia, jakim jest Sprint**. Wydarzenia te są konstruktywne i produktywne, ponieważ tworzą filary empiryzmu w Scrumie, czyli przejrzystość, inspekcję i adaptację.

## Ważne informacje

- Wszystkie wydarzenia są ograniczone czasowo.
- Są ukierunkowane na dostarczenie przyrostu produktu



# Co to jest sprint?

- To czas, kiedy **zespół będzie tworzył kolejny przyrost produktu.**

Sprint zawiera w sobie wszystkie inne wydarzenia i rozpoczyna się od Planowania Sprintu, a kończy Retrospektywą Sprintu. Każda Sprint posiada swój Cel, ustalany przy planowaniu. W trakcie Sprintu Zespół wykonuje **wszystkie zadania niezbędne do osiągnięcia tego celu**

- Trwa od tygodnia do czterech tygodni.
- Długość sprintu powinna być dopasowana do możliwości realizacji celu sprintu.



# Planowanie Sprintu

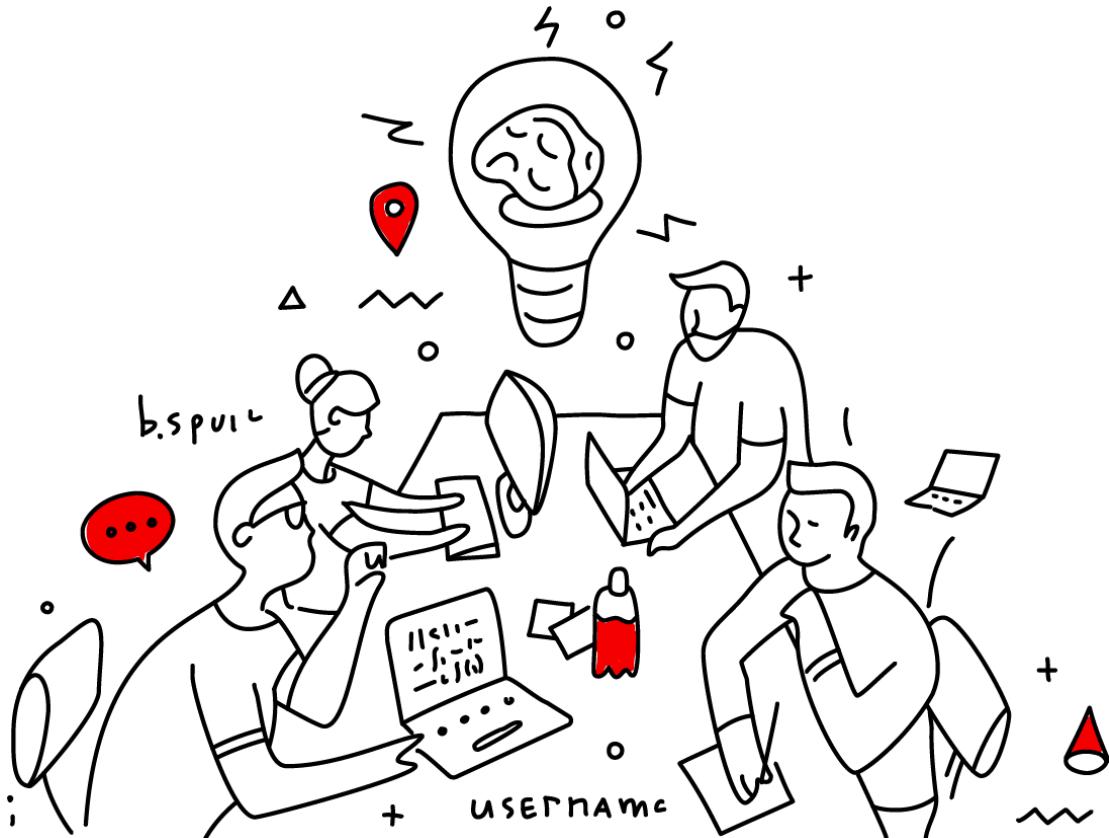
**Planowanie Sprintu (Sprint Planning) – służy do ustalenia celu i zakresu Sprintu.**

W trakcie pierwszej ustala się "po co" i "co" jest do zrobienia. Z Backlogu Produktu wybierane są najważniejsze elementy mające przybliżyć zespół do osiągnięcia celu produktu.

- Pierwsza część planowania poświęcona jest dyskusji nad celem biznesowym (po co i co).
- Druga część służy natomiast do dyskusji "Jak" zostanie wykonana ta praca.
- Zespół powinien skończyć to spotkanie z realnym planem na obecny Sprint.
- Maksymalnie 4 godziny.



# Daily Scrum



- To codzienne spotkanie zespołu mające na celu **synchronizację pracy** i sprawdzenie postępu w kierunku osiągnięcia celu sprintu.
- Szczegóły poszczególnych zagadnień powinny być omówione w wąskim gronie poza Daily
- **Daily jest dla zespołu.** Na Daily może pojawić się dowolna osoba, która jest zainteresowana tym co się dzieje. Natomiast jej rola ogranicza się do bycia obserwatorem.
- Daily nie jest podsumowaniem prac i omówieniem poszczególnych tasków.

# Sprint Review



- Najczęściej kojarzy się z **zaprezentowaniem interesariuszom działającej funkcjonalności**, zebraniem od nich informacji zwrotnej oraz dyskusją na temat możliwości dalszego rozwoju produktu.
- Nie może to być samo „demo”, ale zebranie informacji zwrotnej i dyskusja na temat dalszego rozwoju produktu. Potrzebny jest aktywny udział PO oraz interesariuszy.
- Wynikiem spotkania powinna być aktualizacja backlogu produktu.
- Maksymalnie 2 godziny.

# Retrospekywa



- Spotkanie kończące Sprint, poświęcone sposobie pracy.
- Maksymalnie 1,5h.
- Dotyczy procesu, a nie konkretnych osób.
- Nie ma jednego uniwersalnego modelu retrospekywy.

# Porządkowanie backlogu

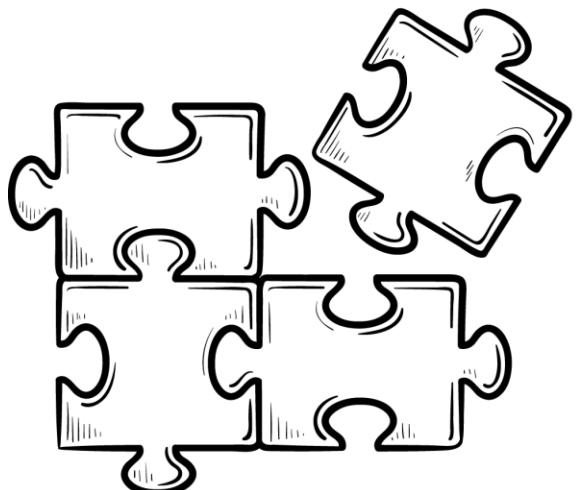


- Dzieje się w trakcie trwania Sprintu. **Polega na porządkowaniu i uszczegóławianiu zadań**, które najprawdopodobniej znajdą się w następnej Iteracji(sprincie), tak aby były gotowe na Planowanie Sprintu.
- Zajmuje do 10% czasu sprintu. Nie musi być to dedykowane spotkanie.
- Jest to wspólną odpowiedzialnością Product Owner i Zespołu.
- W takich spotkaniach często uczestniczą też różni eksperci, klienci i użytkownicy.

# Artefakty



## INCREMENT



## PRODUCT

- Jest tym co Zespół powinien dostarczyć na koniec Sprintu, czyli działającym produktem rozszerzonym o nowe możliwości (funkcjonalności) w stosunku do wersji z poprzedniego Sprintu.
- Zobowiązaniem dla Przyrostu jest **Definicja Ukończenia (Definition of Done, DoD)**, czyli lista kryteriów jakościowych, które musi spełniać przyrost produktu, żeby móc powiedzieć, że zespół skończył nad nim pracę.

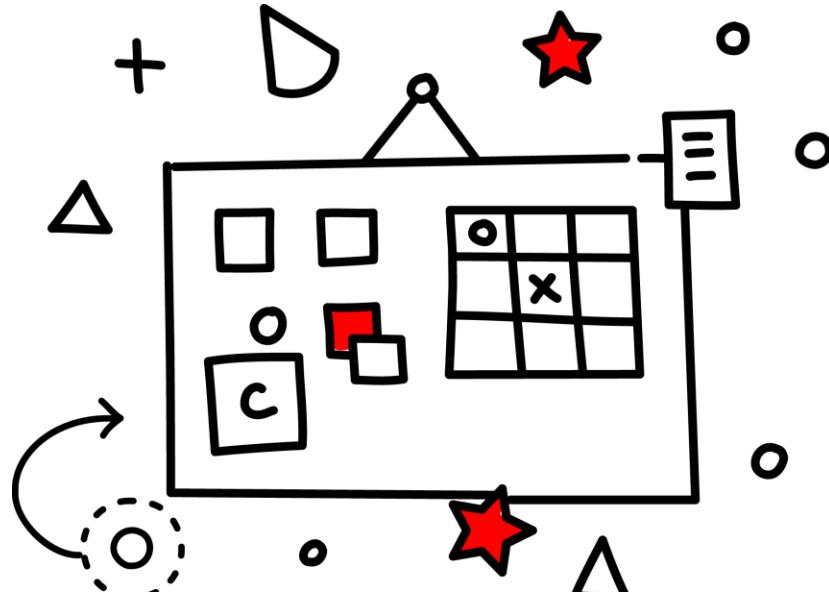
# Backlog produktu



**PRODUCT BACKLOG**

- Uszeregowana lista potrzeb/wymagań.
- Zobowiązaniem dla Backlogu Produktu jest **cel produktu**, czyli co Product Owner chce osiągnąć w najbliższym czasie.
- Product Owner jest odpowiedzialny za zarządzanie tą listą, natomiast zespół za dostarczenie jej części w kolejnych Sprintach.

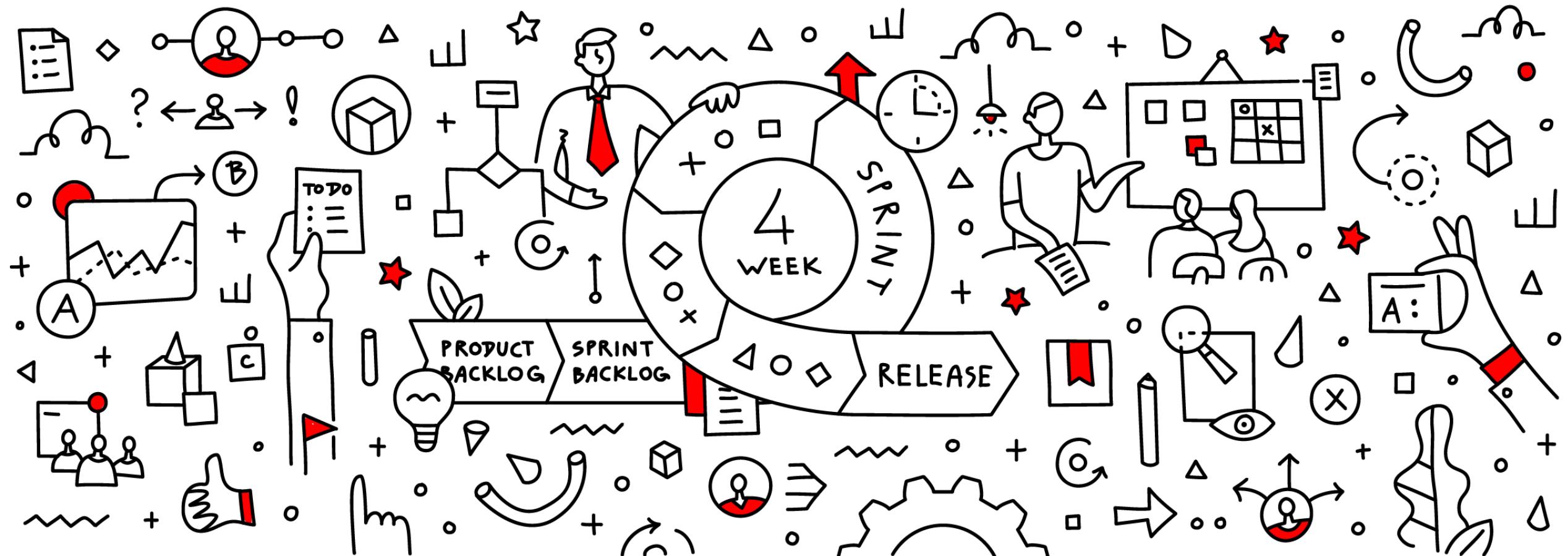
# Backlog sprintu



## SPRINT BACKLOG

- Lista elementów wybranych z Backlogu Produktu do Sprintu, czyli na przykład historyjki, które chcemy zrobić w trakcie Sprintu, oraz plan jak je chcemy zrobić.
- Zobowiązaniem dla Backlogu Sprintu jest **cel sprintu** pozwalający się skupić całemu zespołowi na tym co jest najważniejsze w danej iteracji.

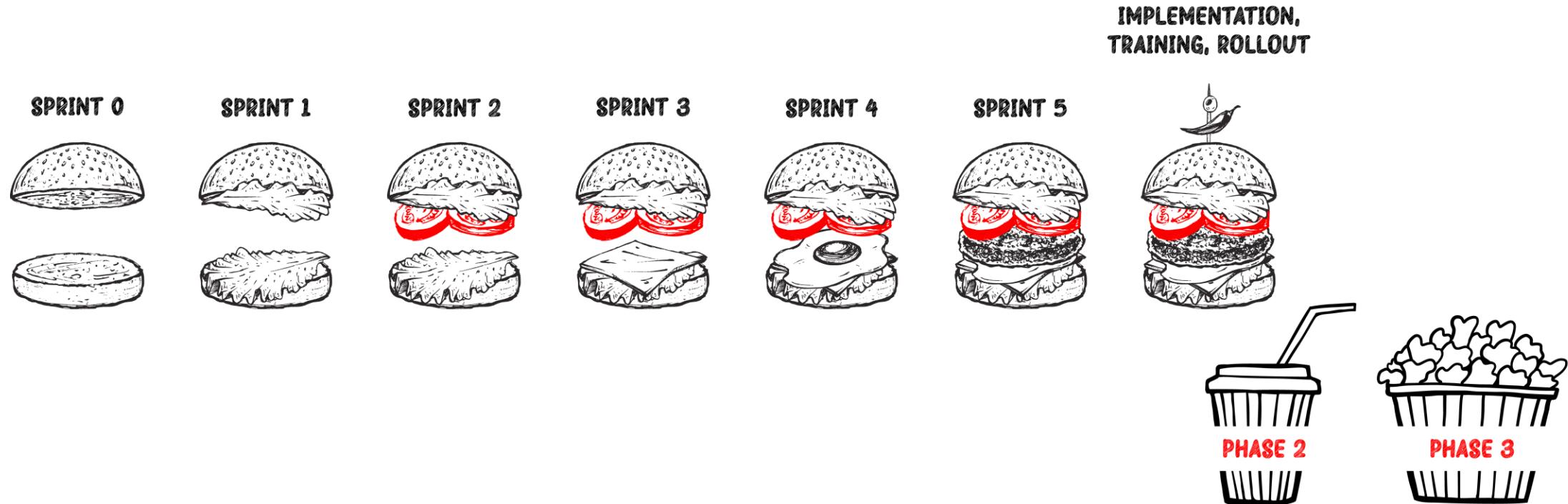
# Przyrost produktu



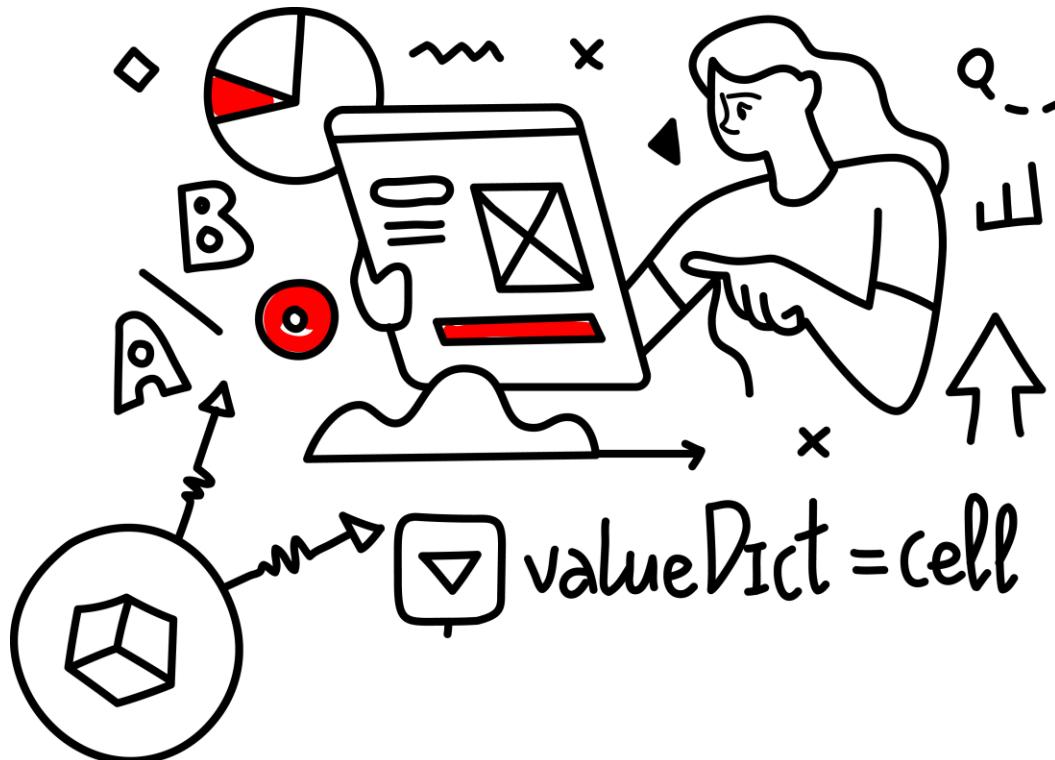
# Inkrementacja



Praca nad fragmentami projektu, które są możliwe do wdrożenia oraz niosą ze sobą wartość biznesową.

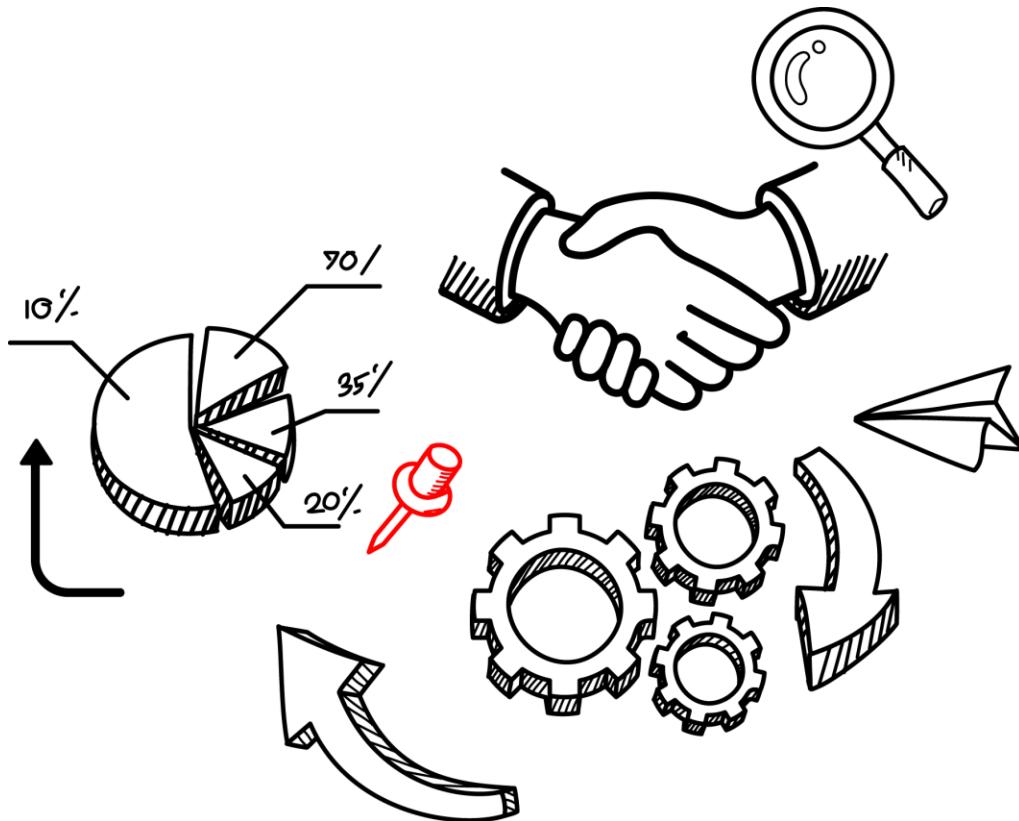


# Definition of Done (DOD)



- Ma zapewnić przejrzystość tego, co to znaczy „ukończona praca”. Dotyczy jednego, konkretnego zespołu pracującego nad konkretnym produktem.
- Są to zasady wypracowywane wspólnie przez zespół, a nie narzucone z zewnątrz. Jednak jeśli organizacja formułuje pewne ograniczenia, oczekiwania, czy standardy dla zespołów, to powinny one przyjąć te oczekiwania jako część swojego Definition of Done.
- Definicja Ukończenia tworzy przejrzystość. Powoduje, że wszyscy tak samo rozumieją, jaką pracę trzeba wykonać, żeby osiągnąć przyrost, który będzie ukończony.

# Definition of Ready



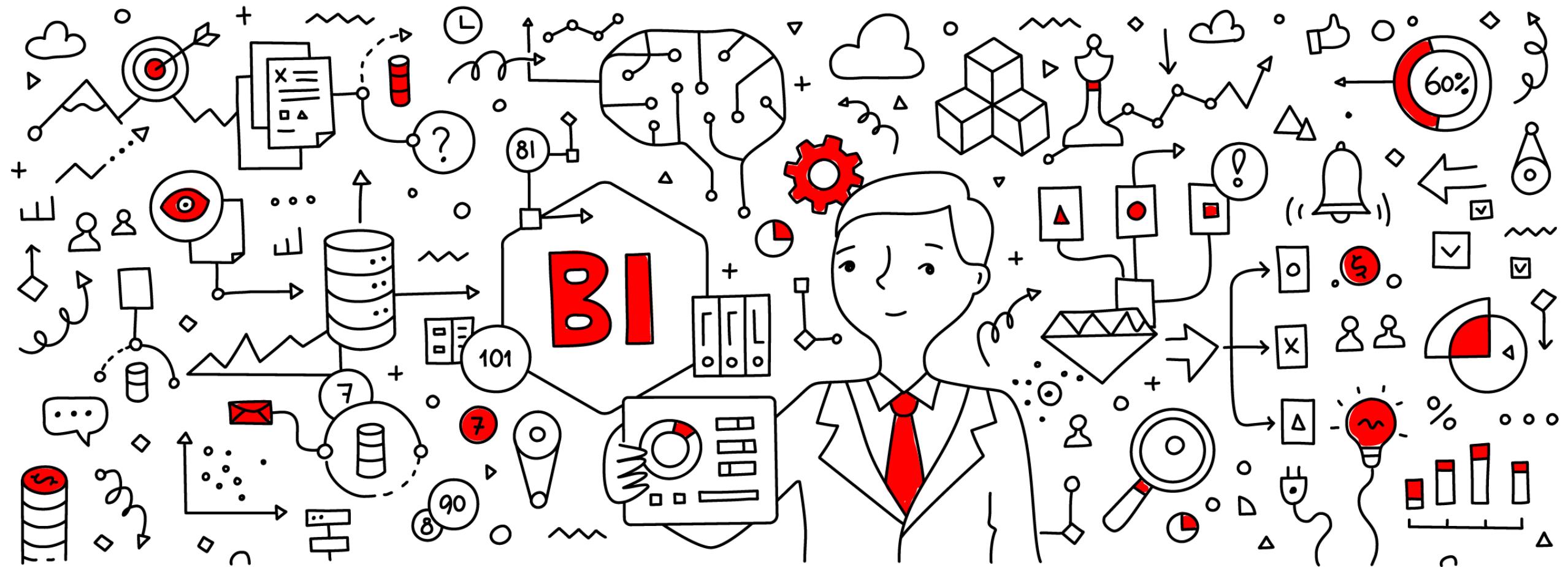
- Określa, kiedy zadanie można uznać za gotowe do przyjęcia do realizacji. Jego celem jest "odsianie" zadań źle lub niejasno zdefiniowanych.
- Działa na poziomie zespołu, a nie organizacji.
- Pozwala jasno określić, że znamy i rozumiemy zakres końcowy zadania .

# Kryteria Akceptacji



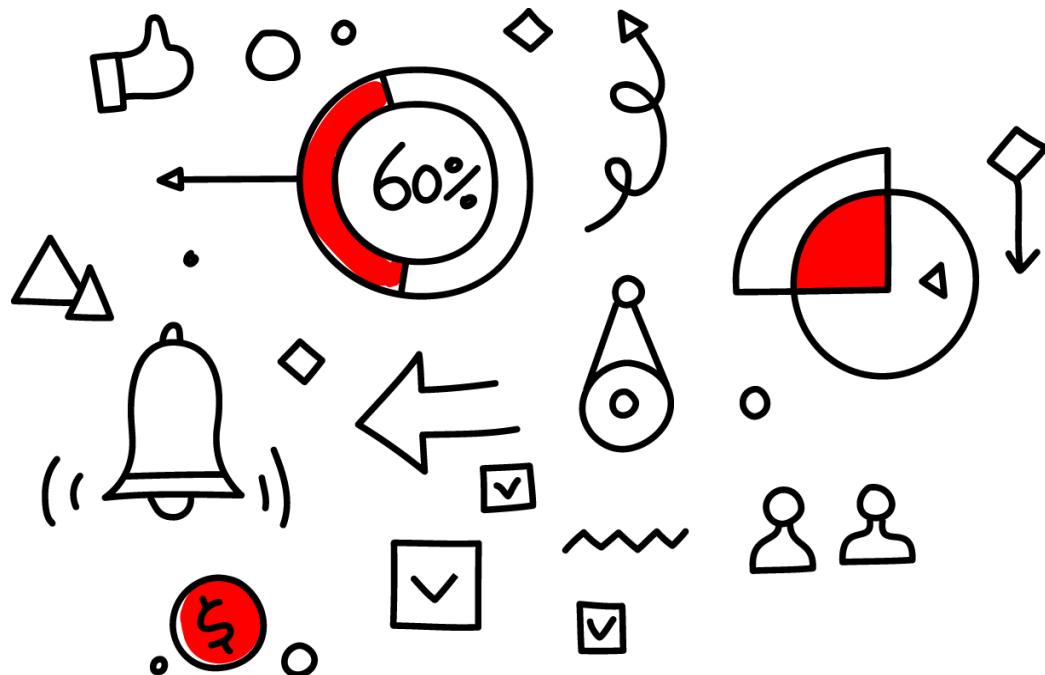
- Lista konkretnych wymagań i warunków do spełnienia, aby Zespół Scrumowy wiedział, czy historyjka została poprawnie wykonana z biznesowego punktu widzenia.
- Każda historyjka powinna mieć swoje kryterium akceptacji.
- Kryteria powinny zawsze opisywać tylko to, CO ma zostać wykonane, a nie za pomocą jakich rozwiązań dojdziemy do efektu końcowego.
- Powinny się pojawić na etapie tworzenia historyjki.
- Są tworzone per historyjka, nie są uniwersalne.

# Szacowanie



# Estymowanie relatywne

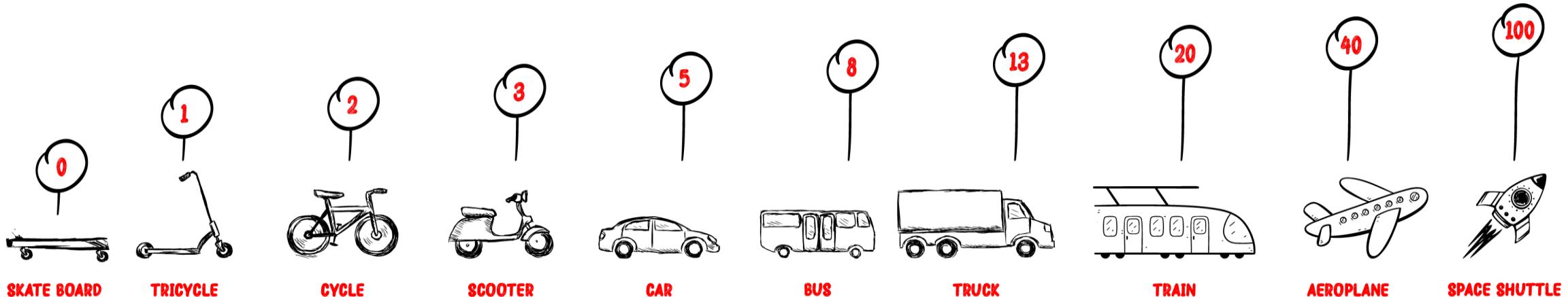
Estymowanie jest to proces znalezienia wyceny lub jej ekstrapolacji, czasami bazujący na danych niekompletnych, nieprzewidywalnych, niestabilnych. Estymowanie mówi nam, jak wiele pieniędzy, wysiłku, zasobów i czasu poświęcimy na zbudowanie określonego systemu lub produktu przy określonych ryzykach.



- Dokonując wyceny elementów backlogu zespół powinien wziąć pod uwagę następujące czynniki: ryzyko/niepewność, złożoność oraz nakład pracy.
- Im bardziej złożone wymaganie, tym większy nakład pracy oraz większe ryzyko i niepewność związane z jego realizacją
- Rodzaje: story points, t-shirt size.

# Story Points

- Do szacowania przy wykorzystaniu miary względnej, jak np. Story Points, wykorzystywany jest najczęściej **ciąg Fibonacciego**.
- (0), 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 ...
- Im większe wymaganie i im mniej wiemy o koniecznych do implementacji funkcjonalnościach, tym nasza wycena posiada większy margines.



# User Stories



# User Stories



- Historyjki użytkowników są zadaniami programistycznymi, które często mają formę „persona + potrzeba + cel”.
- Nadrzędnym założeniem jest to, że absolutnie każde sensowne wymaganie da się zapisać w przedstawiony powyżej sposób.
- Po zapoznaniu się z historyjką użytkownika zespół wie, dlaczego tworzy, co tak właściwie tworzy i jakie korzyści zapewni wynik ich pracy.
- Więcej na wykładzie dotyczącym podstaw pracy z wymaganiami.

# Schemat historyjki

„Jako [kto] [chcę], [w celu]”.



## Kto?

Kto będzie używać danej funkcjonalności ?

Każdy feature może być używany przez kilka osób, chcemy poznać perspektywę każdej.



## Chcę

Opisujemy jego zamiary, a nie funkcje, z których korzysta.

Co próbuje tak naprawdę osiągnąć?



## W celu

Jakie ogólne korzyści próbuje osiągnąć? Jaki wygląda problem wymagający rozwiązania?

# Bibliografia



- Scrum i nie tylko. Teoria i praktyka w metodach Agile, K.Kaczor, PWN, 2016
- Labirynty Scruma, J. Wieczorek, Jacek Wieczorek Consulting 2017
- Tworzenie oprogramowania w 30 dni. Programuj zwinnie i szybko!  
Ken Schwaber, Jeff Sutherland, Helion 2016



WP

DZIĘKUJEMY!

