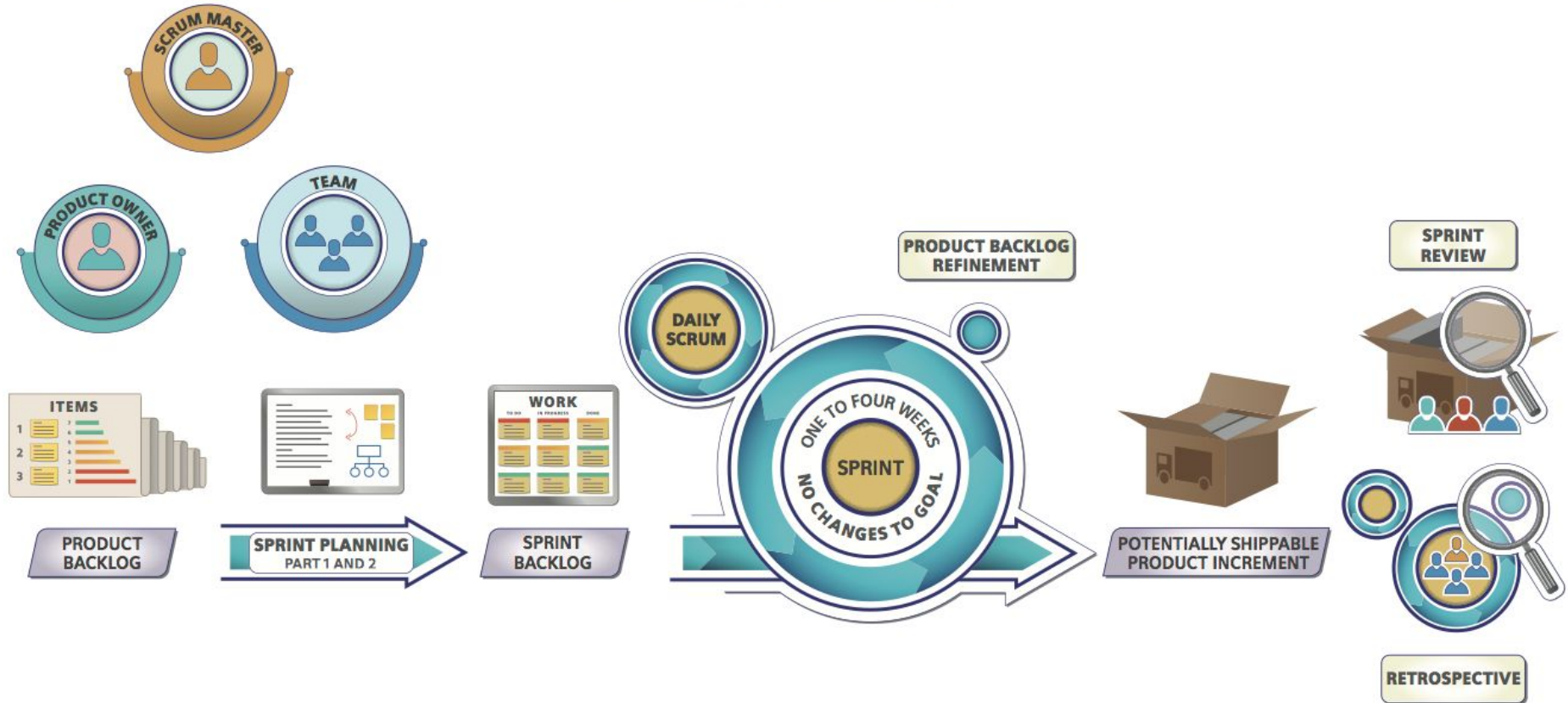


# Large-scale Scrum - LeSS



... oraz trochę o SAFe i Nexus

# SCRUM



**W takim razie czym jest LeSS?**

## **W takim razie czym jest LeSS?**

“LeSS is Scrum applied to many teams working together on one product.” — Oficjalna strona LeSS

**Co to znaczy?**

**Co to znaczy?**

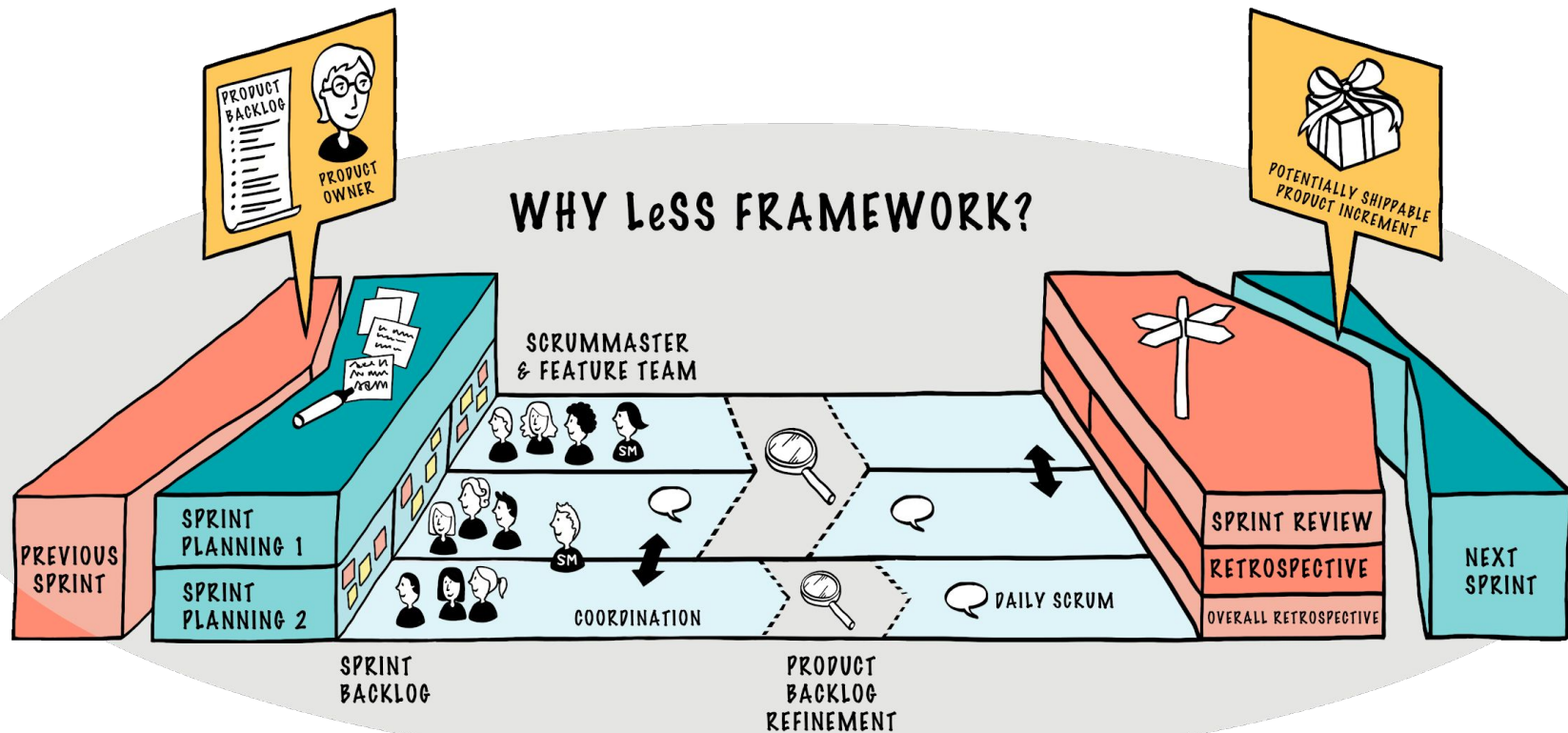
LeSS to przeskalowany Scrum

## Co to znaczy?

LeSS to przeskalowany Scrum

Pracujemy w Scrumie nad **jednym produktem** ale w **wielu zespołach**

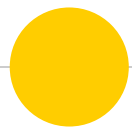
# WHY LESS FRAMEWORK?







ONE  
SCRUM  
TO  
RULE  
THEM  
ALL



# Zasady Less

Sztywne reguły, których powinniśmy się trzymać

# **Zasady struktury**

## **Zasady struktury**

- Zespół jest podstawową częścią budującą organizację

## Zasady struktury

- Zespół jest podstawową częścią budującą organizację
- **Każdy zespół jest samzarządzający, interdyscyplinarny, skupiony w jednym miejscu, i długowieczny(?)**

## Zasady struktury

- Zespół jest podstawową częścią budującą organizację
- Każdy zespół jest samzarządzający, interdyscyplinarny, skupiony w jednym miejscu, i długowieczny(?)
- **Scrum Masterzy są odpowiedzialni za poprawną implementację LeSS**

## Zasady struktury

- Zespół jest podstawową częścią budującą organizację
- Każdy zespół jest samzarządzający, interdyscyplinarny, skupiony w jednym miejscu, i długowieczny(?)
- Scrum Masterzy są odpowiedzialni za poprawną implementację LeSS
- **Scrum Master to pełnoprawna rola w organizacji**

## Zasady struktury

- Zespół jest podstawową częścią budującą organizację
- Każdy zespół jest samzarządzający, interdyscyplinarny, skupiony w jednym miejscu, i długowieczny(?)
- Scrum Masterzy są odpowiedzialni za poprawną implementację LeSS
- Scrum Master to pełnoprawna rola w organizacji
- **Jeden Scrum Master może pomagać 1-3 zespołom**



# Zasady produktu

# Zasady produktu

- Jeden Product Backlog i Product Owner dla każdego produktu

# Zasady produktu

- Jeden Product Backlog i Product Owner dla każdego produktu
- **Product Owner powinien współpracować z zespołami przy tworzeniu Product Backlogu**

# Zasady produktu

- Jeden Product Backlog i Product Owner dla każdego produktu
- Product Owner powinien współpracować z zespołami przy tworzeniu Product Backlogu
- **Nadawanie priorytetów dla elementów w kolejce powinno zostać uzgodnione z Product Ownerem**

# Zasady produktu

- Jeden Product Backlog i Product Owner dla każdego produktu
- Product Owner powinien współpracować z zespołami przy tworzeniu Product Backlogu
- Nadawanie priorytetów dla elementów w kolejce powinno zostać uzgodnione z Product Ownerem
- **Wyjaśnianie szczegółów dotyczących zadań może zostać uzgadniane między zespołami i klientem**

# Zasady produktu

- Jeden Product Backlog i Product Owner dla każdego produktu
- Product Owner powinien współpracować z zespołami przy tworzeniu Product Backlogu
- Nadawanie priorytetów dla elementów w kolejce powinno zostać uzgodnione z Product Ownerem
- Wyjaśnianie szczegółów dotyczących zadań może zostać uzgadniane między zespołami i klientem
- **Jedno Definition of Done(DoD) produktu wspólne dla wszystkich zespołów**

# Zasady produktu

- Jeden Product Backlog i Product Owner dla każdego produktu
- Product Owner powinien współpracować z zespołami przy tworzeniu Product Backlogu
- Nadawanie priorytetów dla elementów w kolejce powinno zostać uzgodnione z Product Ownerem
- Wyjaśnianie szczegółów dotyczących zadań może zostać uzgadniane między zespołami i klientem
- Jedno Definition of Done(DoD) produktu wspólne dla wszystkich zespołów
- **Każdy zespół może mieć bardziej doprecyzowane Definition of Done rozwijając istniejące rozwiązanie**

# Zasady produktu

- Jeden Product Backlog i Product Owner dla każdego produktu
- Product Owner powinien współpracować z zespołami przy tworzeniu Product Backlogu
- Nadawanie priorytetów dla elementów w kolejce powinno zostać uzgodnione z Product Ownerem
- Wyjaśnianie szczegółów dotyczących zadań może zostać uzgadniane między zespołami i klientem
- Jedno Definition of Done(DoD) produktu wspólne dla wszystkich zespołów
- Każdy zespół może mieć bardziej doprecyzowane Definition of Done rozwijając istniejące rozwiązanie
- **DoD powinno pozwolić wytworzyć produkt, który możemy wdrożyć po każdym sprincie**



# Zasady Sprintu

# Zasady Sprintu

- Istnieje tylko jeden Sprint na poziomie Produktu

# Zasady Sprintu

- Istnieje tylko jeden Sprint na poziomie Produktu
- **Sprint Planning(SP)** składa się z dwóch części: **One i Two**

# Zasady Sprintu

- Istnieje tylko jeden Sprint na poziomie Produktu
- Sprint Planning(SP) składa się z dwóch części: One i Two
- **Podczas SP One zespoły lub ich przedstawiciele ustalają nad czym będą pracować podczas sprintu**

# Zasady Sprintu

- Istnieje tylko jeden Sprint na poziomie Produktu
- Sprint Planning(SP) składa się z dwóch części: One i Two
- Podczas SP One zespoły lub ich przedstawiciele ustalają nad czym będą pracować podczas sprintu
- **Podczas SP Two zespoły wewnętrznie ustalają co będzie w Sprint Backlogu**

# Zasady Sprintu

- Istnieje tylko jeden Sprint na poziomie Produktu
- Sprint Planning(SP) składa się z dwóch części: One i Two
- Podczas SP One zespoły lub ich przedstawiciele ustalają nad czym będą pracować podczas sprintu
- Podczas SP Two zespoły wewnętrznie ustalają co będzie w Sprint Backlogu
- **Każdy team odbywa Daily Scrum**

# Zasady Sprintu

- Istnieje tylko jeden Sprint na poziomie Produktu
- Sprint Planning(SP) składa się z dwóch części: One i Two
- Podczas SP One zespoły lub ich przedstawiciele ustalają nad czym będą pracować podczas sprintu
- Podczas SP Two zespoły wewnętrznie ustalają co będzie w Sprint Backlogu
- Każdy team odbywa Daily Scrum
- **Odbywa się jedno Scrum Review wspólne dla wszystkich zespołów**

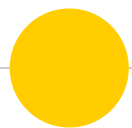
# Zasady Sprintu

- Istnieje tylko jeden Sprint na poziomie Produktu
- Sprint Planning(SP) składa się z dwóch części : One i Two
- Podczas SP One zespoły lub ich przedstawiciele ustalają nad czym będą pracować podczas sprintu
- Podczas SP Two zespoły wewnętrznie ustalają co będzie w Sprint Backlogu
- Każdy team odbywa Daily Scrum
- Odbywa się jedno Scrum Review wspólne dla wszystkich zespołów
- **Każdy Team ma swój własny Sprint Retrospective**



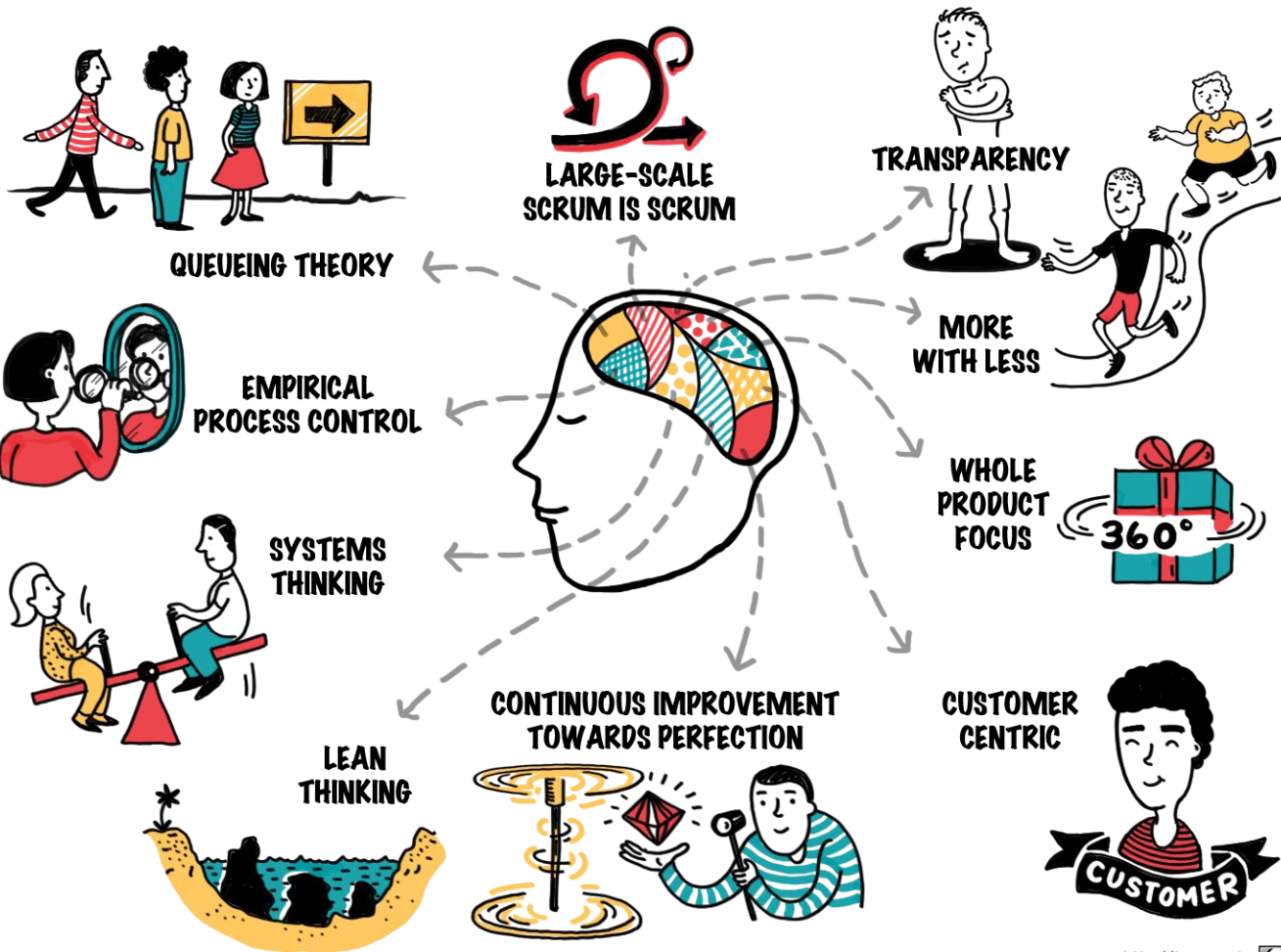
# Zasady Sprintu

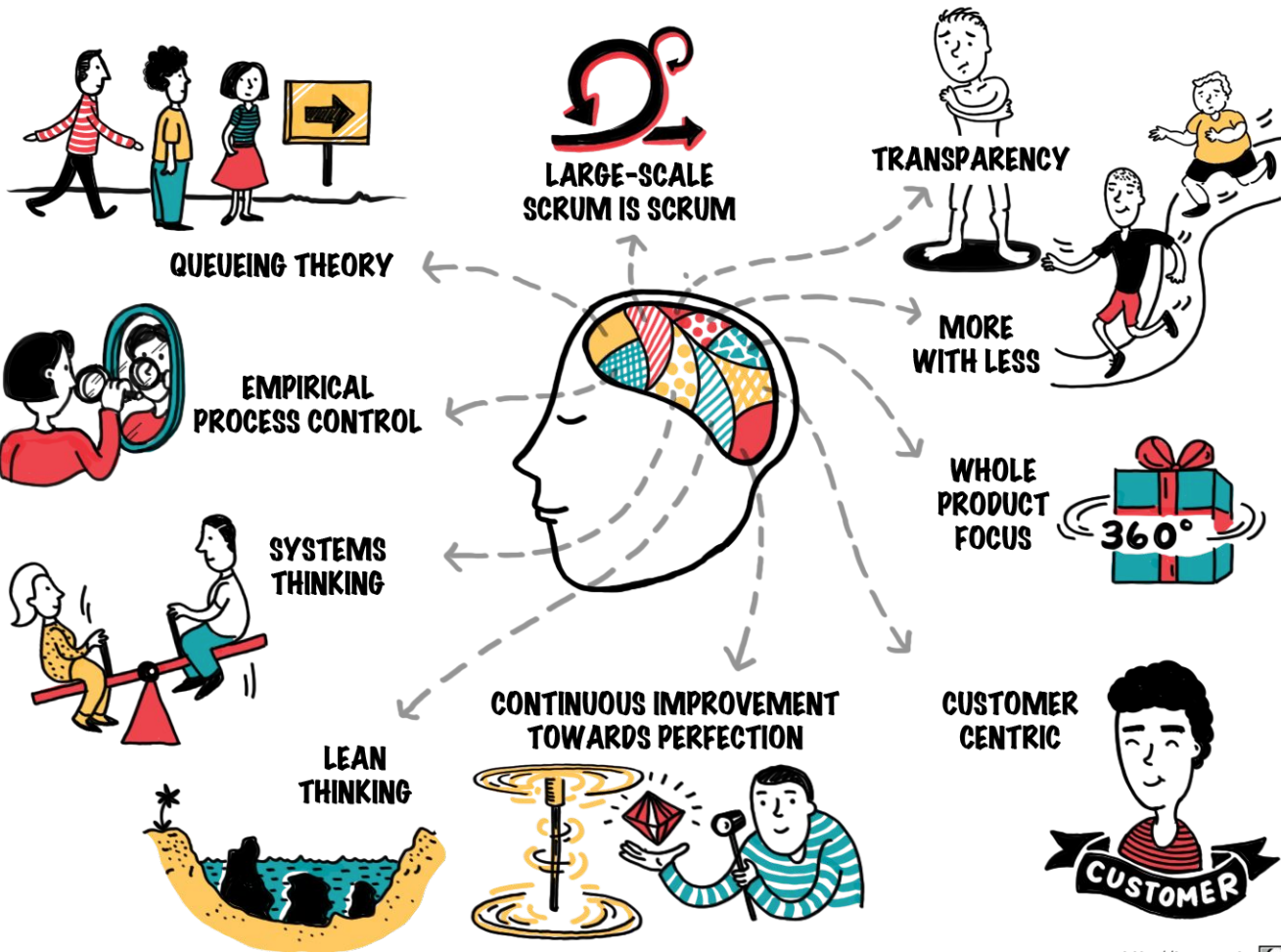
- Istnieje tylko jeden Sprint na poziomie Produktu
- Sprint Planning(SP) składa się z dwóch części: One i Two
- Podczas SP One zespoły lub ich przedstawiciele ustalają nad czym będą pracować podczas sprintu
- Podczas SP Two zespoły wewnętrznie ustalają co będzie w Sprint Backlogu
- Każdy team odbywa Daily Scrum
- Odbywa się jedno Scrum Review wspólne dla wszystkich zespołów
- Każdy Team ma swój własny Sprint Retrospective
- **Product Backlog Refinement odbywa się między zespołami i wewnątrz nich**

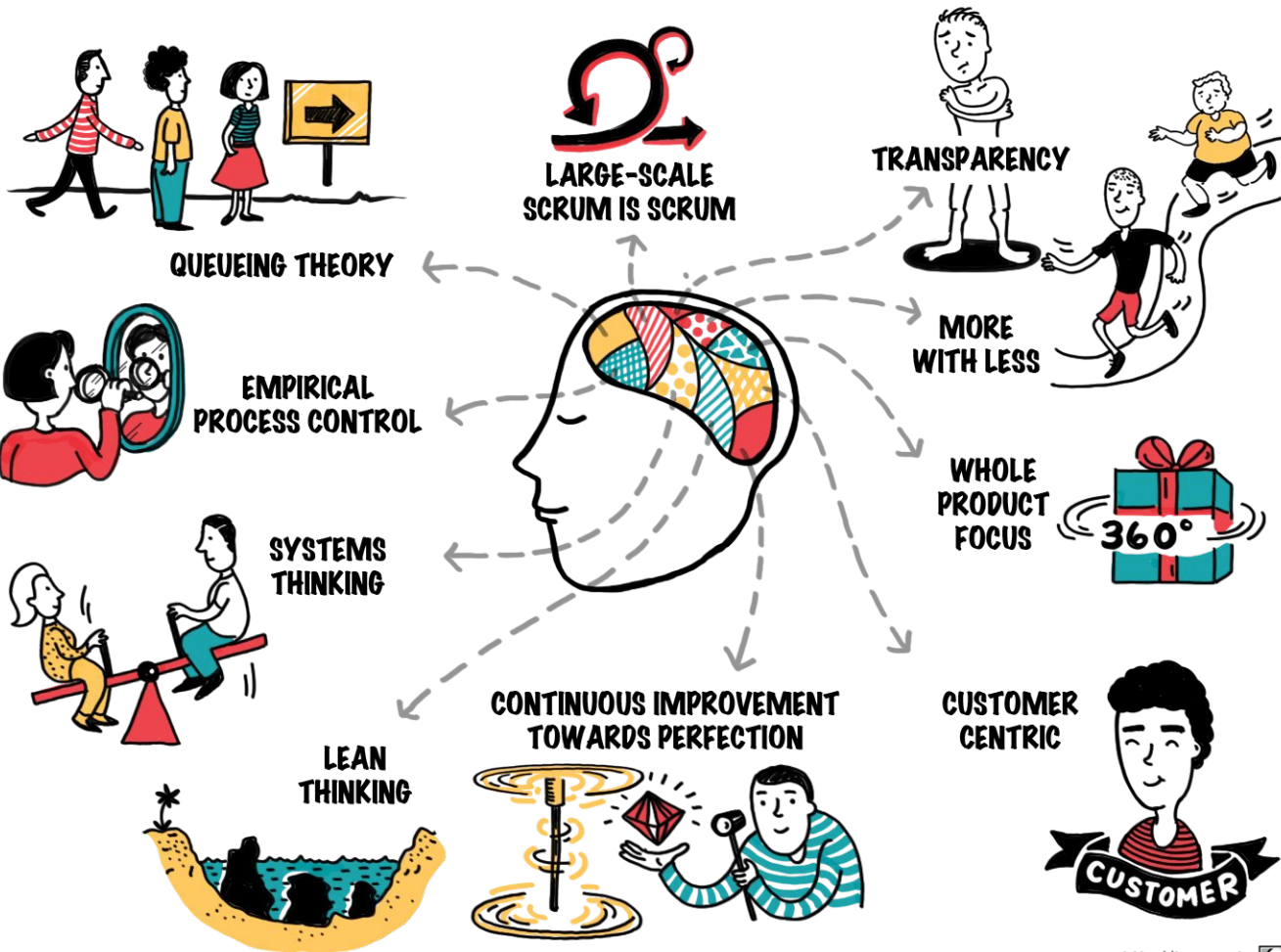


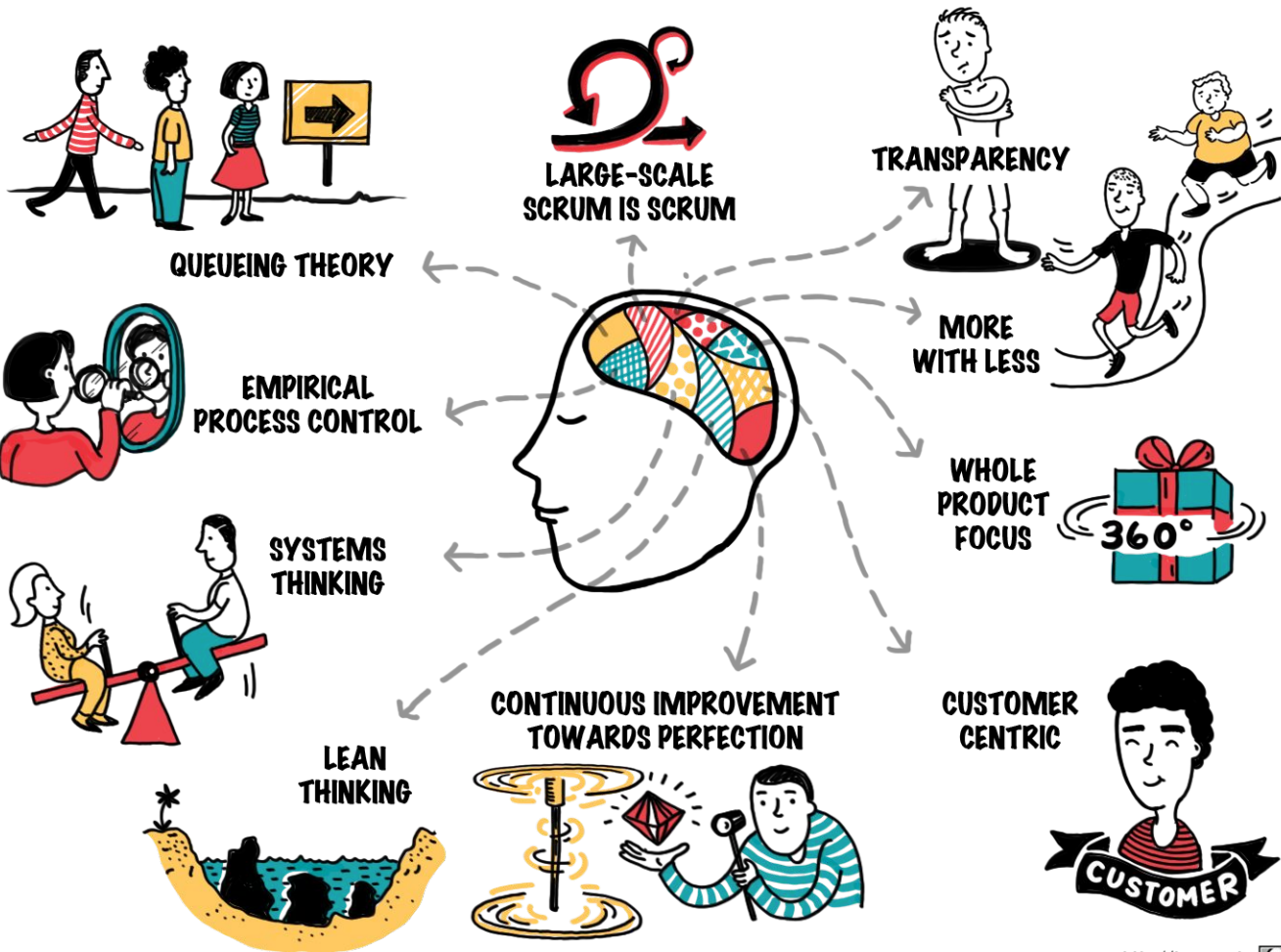
# LeSS Principles

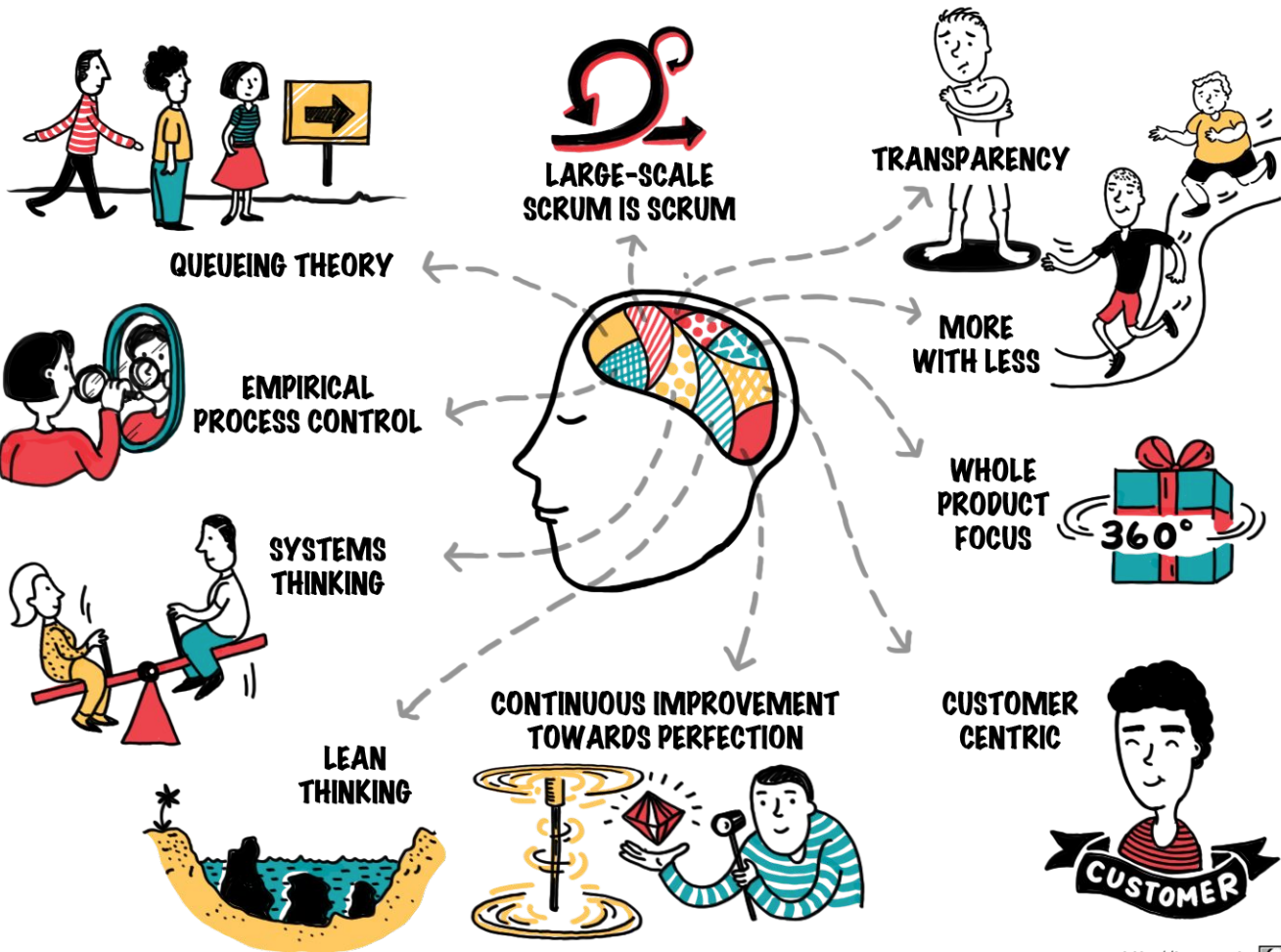
W jaki sposób powinniśmy myśleć implementując LeSS



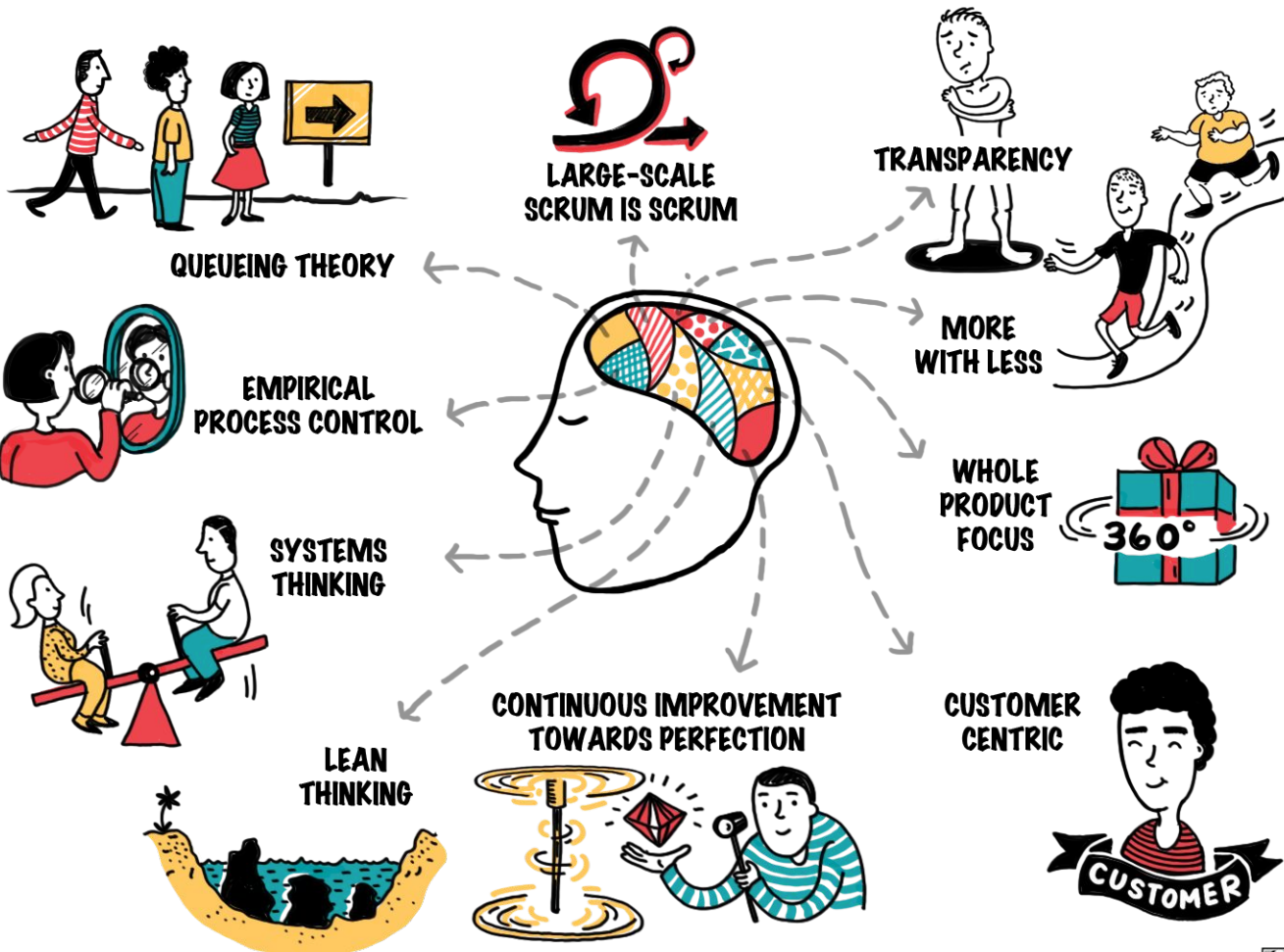




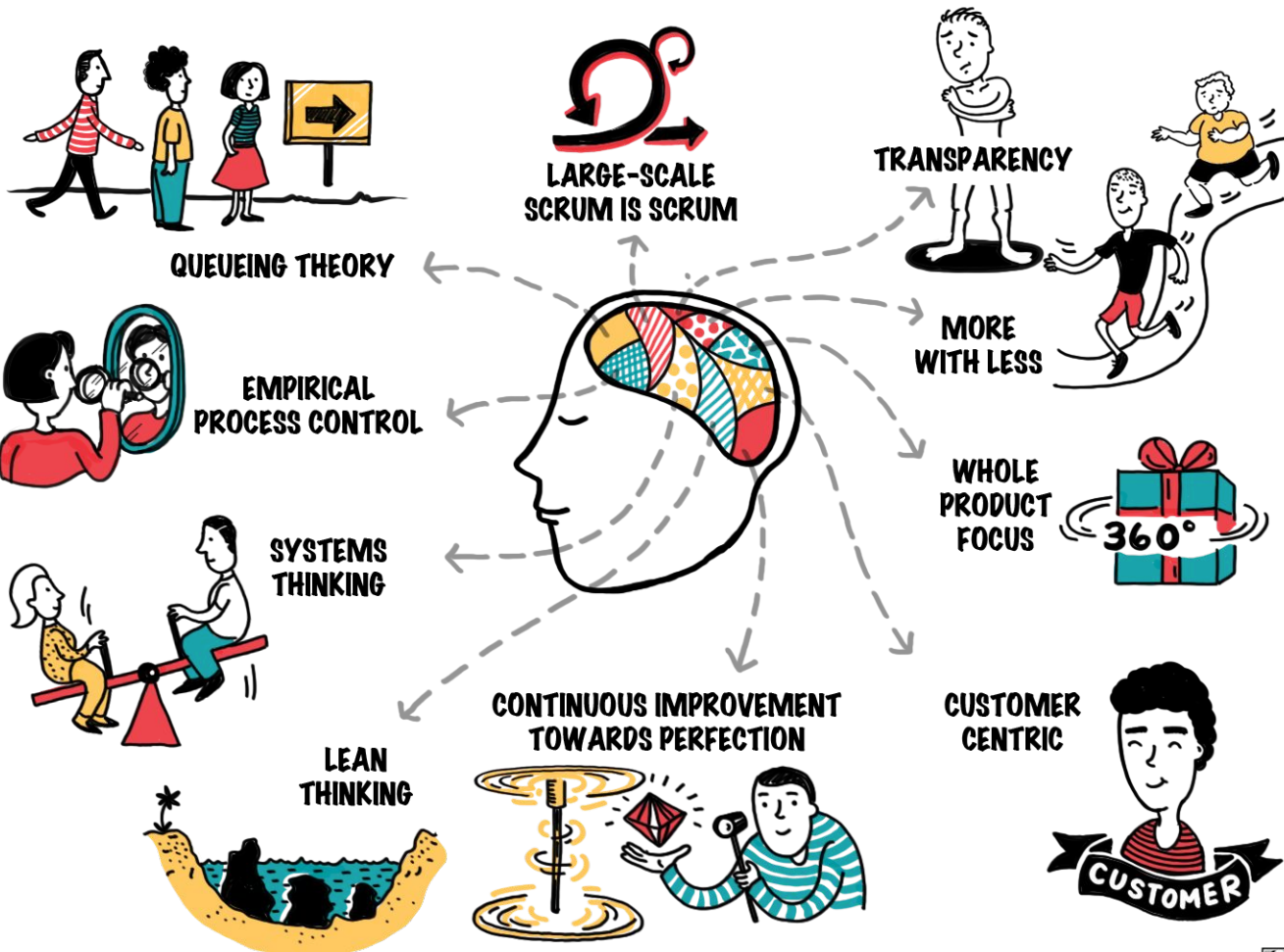


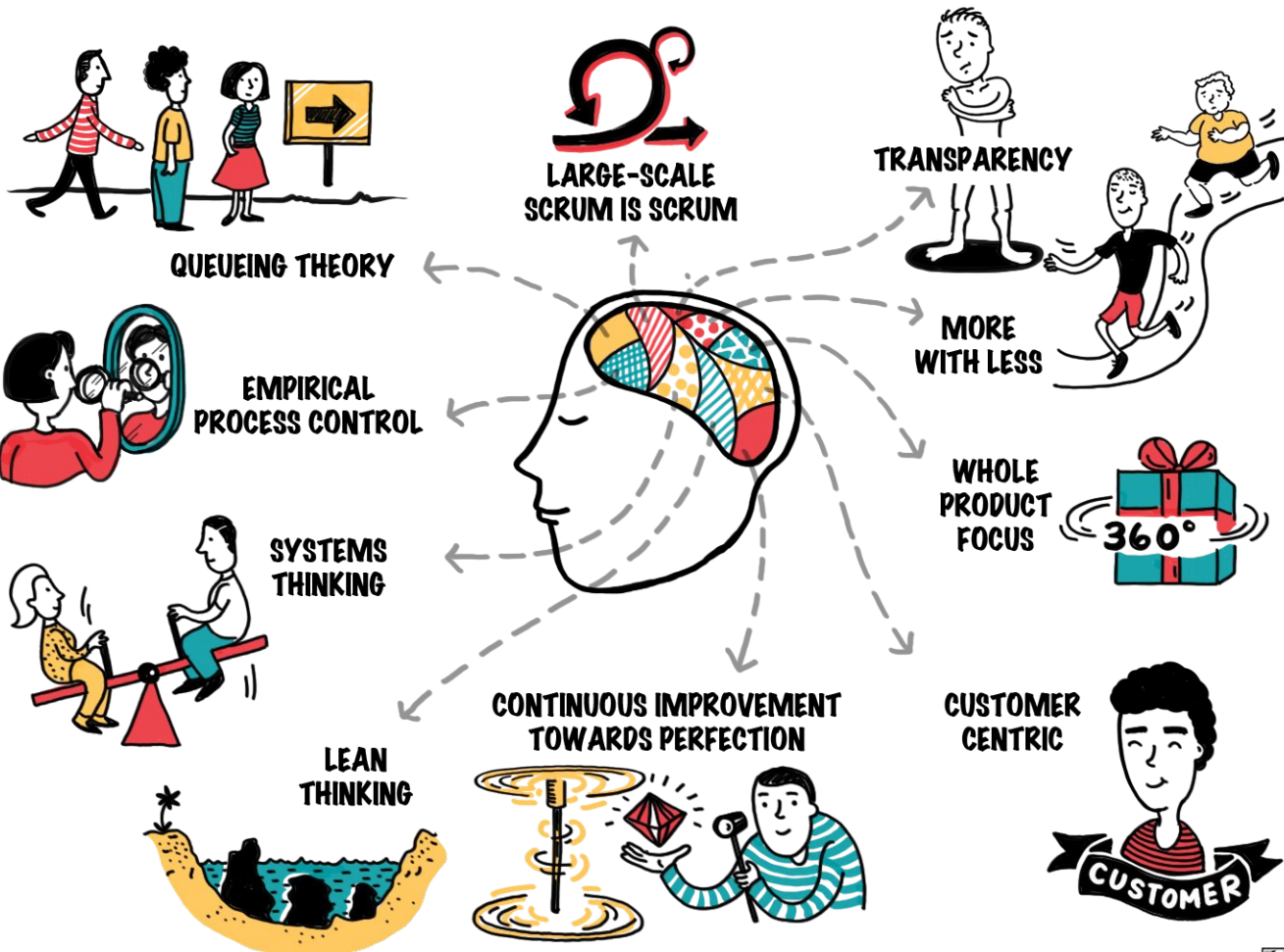


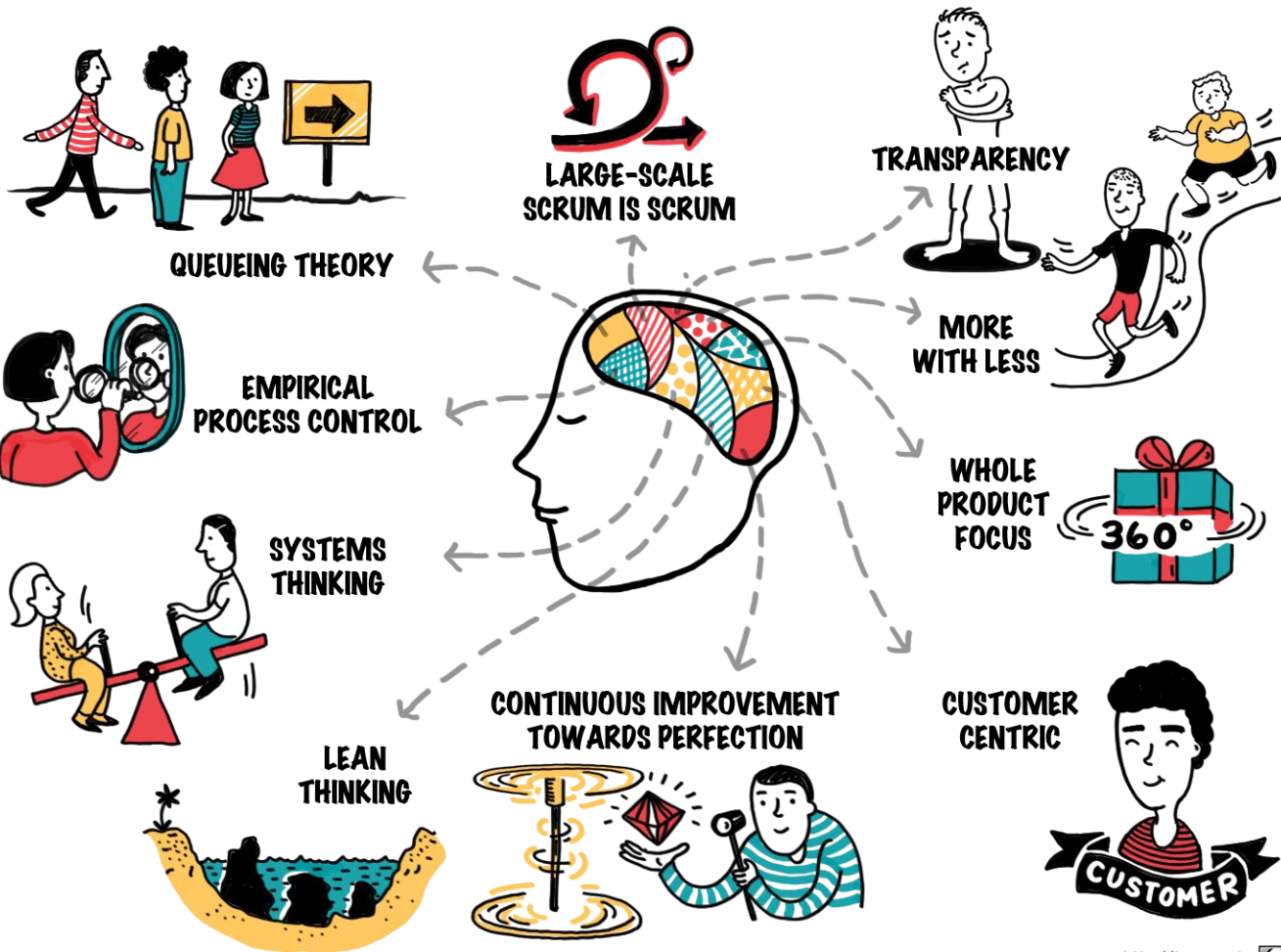


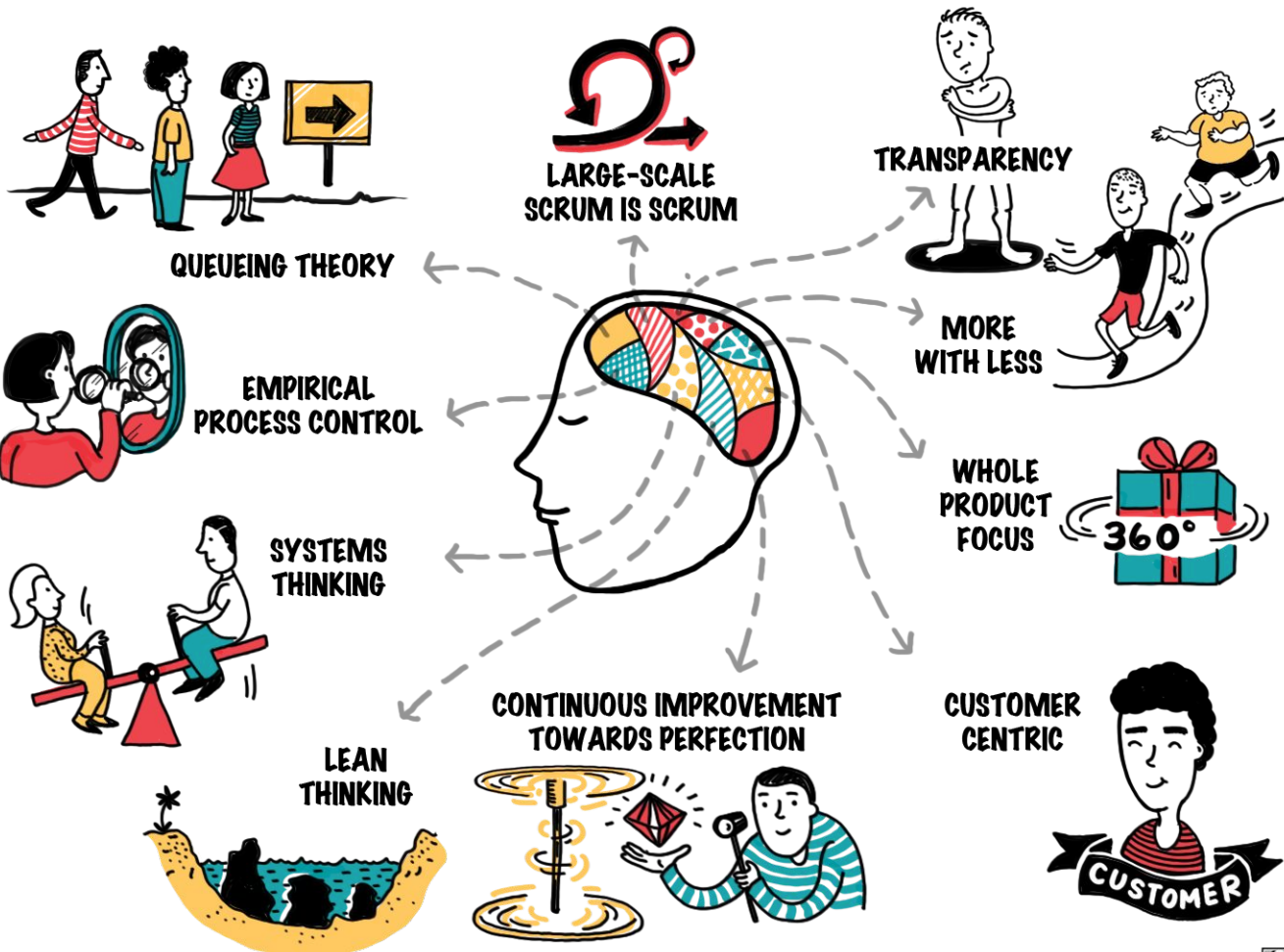


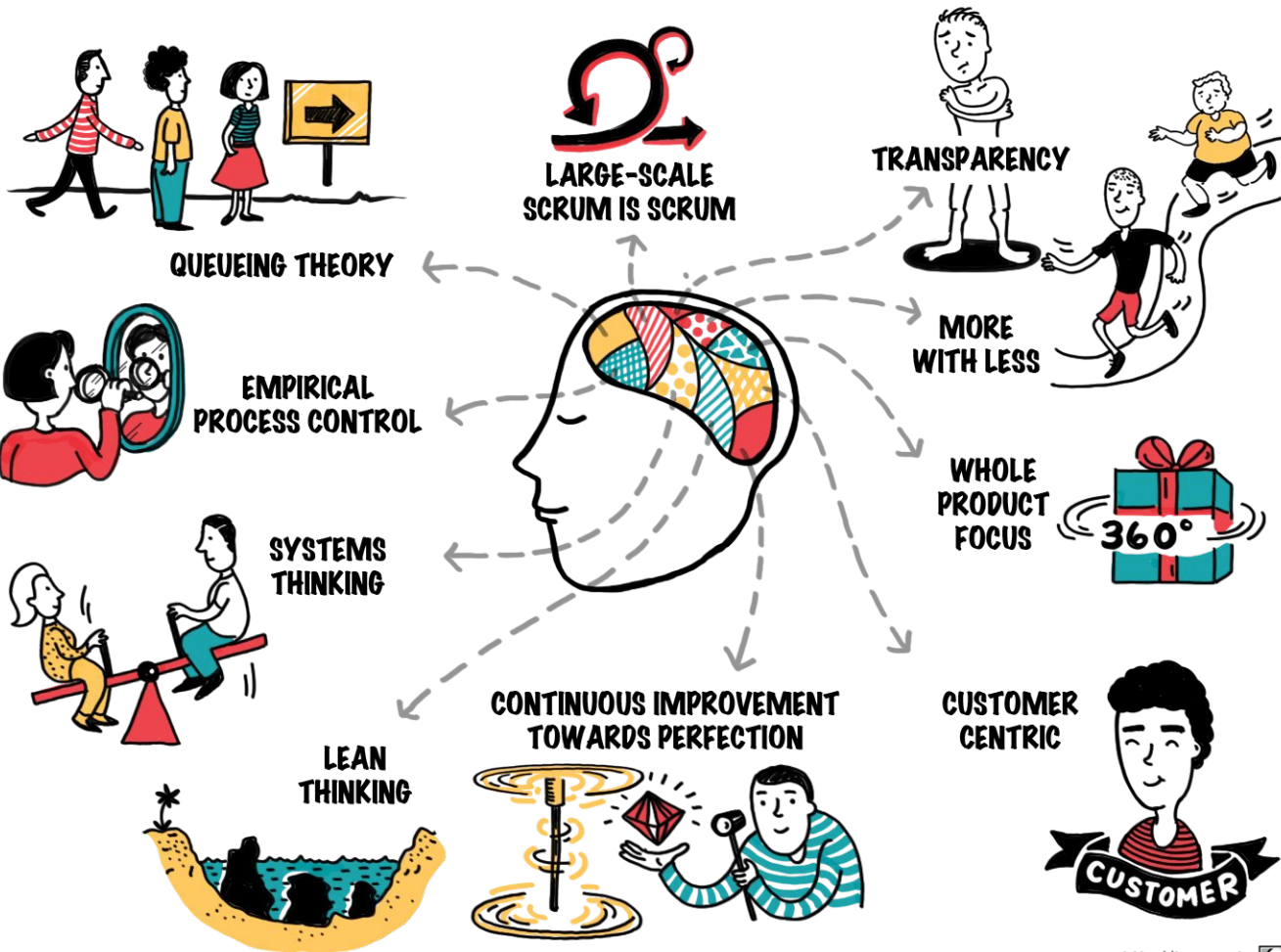














## Omówmy te koncepty

### Large-Scale Scrum is Scrum

Scrum zarządzający wieloma zespołami zamiast wielu zespołów Scrum

### Transparency

Precyzyjne DoD dają łatwy sposób ocenienia, czy dany produkt jest gotowy

### More With Less

Mniej sztywnych procesów, więcej wyciągania wniosków z doświadczeń (w Empirical Process Control)

### Whole Product Focus

Dostarczanie małej części działającego produktu w każdym sprincie, zamiast tworzenie niezintegrowanych funkcjonalności

### Customer Centric

Utrzymywanie dobrej komunikacji z klientem. Świadomość, że nasz kod ma przełożenie na produkt

### Continuous Improvement Towards Perfection

Każda iteracja sprintu powinna być lepiej przemyślana od poprzedniej, na podstawie przeszłych doświadczeń

### **Lean Thinking**

Skupiamy się jedynie na rzeczach, które wnoszą wartość do produktu. Osoby na stanowiskach managerskich powinny chętnie dzielić się wiedzą na temat procesów.

### **Systems thinking**

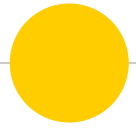
Żaden z zespołów nie powinien być traktowany w lepszy czy gorszy sposób. Każdy z nich pracuje nad integralną częścią aplikacji.

### **Empirical Process Control**

Esencja Scruma, zamiast trzymania się sztywnych reguł powinniśmy uczyć się na błędach i wprowadzać ciągłe poprawki w naszej implementacji tej metodologii.

### **Queueing Theory**

Zrozumienie tej teorii pomoże nam w m.in. tworzeniu lepszych limitów tasków na sprint(?)



# Huge LeSS



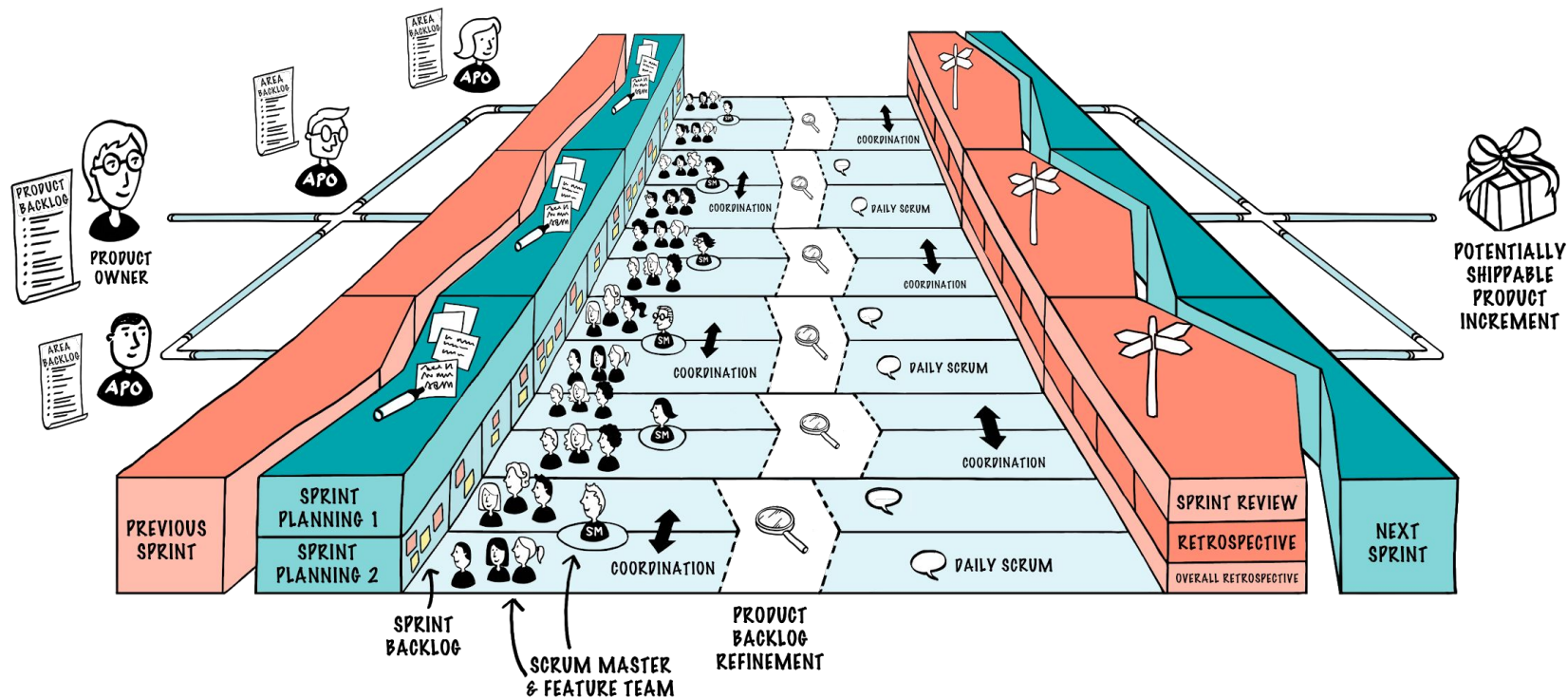
## **Huge LeSS**

Idea jest taka, że idziemy o krok dalej i skalujemy LeSS

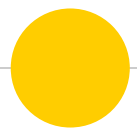
## Huge LeSS

Idea jest taka, że idziemy o krok dalej i skalujemy LeSS

Dodana zostaje rola **Area Product Owner**, czyli Product Owner dla jednego z klastrów(?) LeSS







**Jakie są alternatywy?**

# **Scaled Agile Framework - SAFe**

# **Scaled Agile Framework - SAFe**

To zbiór schematów organizacji ludzi i zadań. Służy do skalowania rozwiązań według zasad Agile i Lean

# **Scaled Agile Framework - SAFe**

To zbiór schematów organizacji ludzi i zadań. Służy do skalowania rozwiązań według zasad Agile i Lean

Najczęściej dzielimy na 3 poziomy: Team, Program i Portfolio(w pełnej wersji zawiera jeszcze jeden poziom Large Solution)



# Scaled Agile Framework - SAFe

To zbiór schematów organizacji ludzi i zadań. Służy do skalowania rozwiązań według zasad Agile i Lean

Najczęściej dzielimy na 3 poziomy: Team, Program i Portfolio(w pełnej wersji zawiera jeszcze jeden poziom Large Solution)

**Team Level:** Kanban, Scrumban, Scrum

# Scaled Agile Framework - SAFe

To zbiór schematów organizacji ludzi i zadań. Służy do skalowania rozwiązań według zasad Agile i Lean

Najczęściej dzielimy na 3 poziomy: Team, Program i Portfolio(w pełnej wersji zawiera jeszcze jeden poziom Large Solution)

**Team Level:** Kanban, Scrumban, Scrum

**Program Level:** Zawiera role i aktywności pozwalające na ciągłe dostarczanie rozwiązań przez Agile Release Train

# Scaled Agile Framework - SAFe

To zbiór schematów organizacji ludzi i zadań. Służy do skalowania rozwiązań według zasad Agile i Lean

Najczęściej dzielimy na 3 poziomy: Team, Program i Portfolio(w pełnej wersji zawiera jeszcze jeden poziom Large Solution)

**Team Level:** Kanban, Scrumban, Scrum

**Program Level:** Zawiera role i aktywności pozwalające na ciągłe dostarczanie rozwiązań przez Agile Release Train

**Portfolio Level:** Zawiera zasady, praktyki i role potrzebne do zarządzania Value Stream

# Scaled Agile Framework - SAFe

To zbiór schematów organizacji ludzi i zadań. Służy do skalowania rozwiązań według zasad Agile i Lean

Najczęściej dzielimy na 3 poziomy: Team, Program i Portfolio(w pełnej wersji zawiera jeszcze jeden poziom Large Solution)

**Team Level:** Kanban, Scrumban, Scrum

**Program Level:** Zawiera role i aktywności pozwalające na ciągłe dostarczanie rozwiązań przez Agile Release Train

**Portfolio Level:** Zawiera zasady, praktyki i role potrzebne do zarządzania Value Stream

Agile Release Train - czyli zbiór wszystkich zespołów

Value Stream - Reprezentuje kroki, które musimy wykonać by stworzyć produkt

**Nexus**

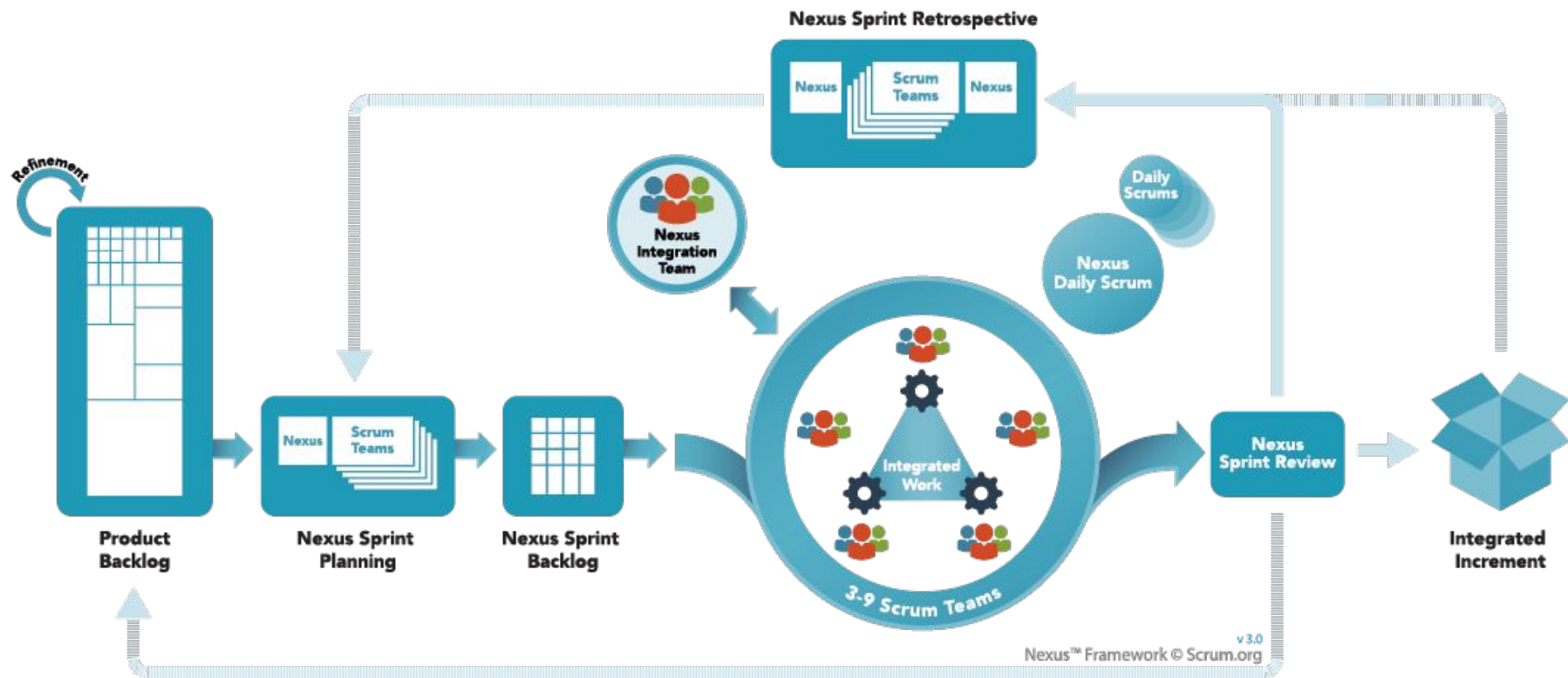
# Nexus

Nexus to framework na który składają się role, zdarzenia, artefakty(?) i zasady, które spajają pracę od 3 do 9 zespołów Scrumowych pracujących nad jednym Product Backlogiem, w celu dostarczania zintegrowanych inkrementacji produktu końcowego.

# Nexus

Nexus to framework na który składają się role, zdarzenia, artefakty(?) i zasady, które spajają pracę od 3 do 9 zespołów Scrumowych pracujących nad jednym Product Backlogiem, w celu dostarczania zintegrowanych inkrementacji produktu końcowego.

Istnieje nowa rola Nexus Integration Team, jest to zespół osób dbających nad integracją pracy wszystkich zespołów.





# Dzięki za uwagę!

## *Macie jakieś pytania ?*

Stworzone przez

● Paweł Bajgrowicz

Jeśli chcecie wiedzieć więcej [LeSS](#), [SAFe](#), [Nexus](#),