LEAN Development

Tom Hoogenboom, System Engineering, ASML

V1 | 6-oct-2015 | Mechelen

Public

LEAN Development *Seen from the other side:*

Tom Hoogenboom, System Engineering, ASML

V1 | 6-oct-2015 | Mechelen

Lean Development

Think big

act small

fail fast

learn rapidly

Source: https://nl.wikipedia.org/wiki/Lean-softwareontwikkeling

Encourage waste

The first idea you have is seldom the best,
wait until idea 52 is there (42 if you are in a hurry).

Lean Development

Encourage waste

The first idea you have is seldom the best,
wait until idea 52 is there (42 if you are in a hurry).

Discourage learning from the past

In SW history does *not* repeat itself.

If you copy, that is what you get – a copy. So don’t:

Always start afresh

Lean Development

Encourage waste

The first idea you have is seldom the best,
wait until idea 52 is there (42 if you are in a hurry).

Discourage learning from the past

In SW history does *not* repeat itself.

If you copy, that is what you get – a copy. So don’t:

Always start afresh

Decide early, using your guts

Let your intuition guide you.
Your first feeling (not your first idea) is usually right.

Decide early, using your guts

Let your intuition guide you.
Your first feeling (not your first idea) is usually right.

Lean Development

Decide early, using your guts

Let your intuition guide you.
Your first feeling (not your first idea) is usually right.

Deliver as late as possible

Surprise your end user with something that actually works.

And be aware that the 10% you missed at your x^{th} RAD delivery will take 90% of the total development cost to integrate at your end user.

Lean Development

Decide early, using your guts

- Let your intuition guide you.
  Your first feeling (not your first idea) is usually right.

Deliver as late as possible

- Surprise your end user with something that actually works.
- And be aware that the 10% you missed at your x^{th} RAD delivery will take 90% of the total development cost to integrate at your end user.

Consider the team as incompetent – but assume you are wrong about that

- Just in case there is a gap – you are responsible for mishaps

Lean Development

Consider the team as incompetent – but assume you are wrong about that

Just in case there is a gap – you are responsible for mishaps

Quality == quality

just strange it is not on slide 1….

Lean Development

Consider the team as incompetent – but assume you are wrong about that

Just in case there is a gap – you are responsible for mishaps

Quality == quality

just strange it is not on slide 1….

Think small

The devil is in the details.