

# PROSES PERANGKAT LUNAK (*SOFTWARE PROCESS*)

Week 02



# DEFINISI MITOS

**mi·tos** *n* 1 cerita suatu bangsa tentang dewa dan pahlawan zaman dahulu, mengandung penafsiran tt asal-usul semesta alam, manusia, dan bangsa tsb mengandung arti mendalam yg diungkapkan dng cara gaib;

**me·mi·tos·kan** *v* mengeramatkan, mengagungkan secara berlebih-lebihan tt pahlawan, benda, dsb; menjadikan mitos; mendewakan: *kecenderungan - seseorang perlu dicegah*;

**pe·mi·tos·an** *n* proses, cara, perbuatan menjadikan mitos; pendewaan

**Mitos - mitos yang ada pada perangkat lunak menyampaikan informasi yang salah dan kebingungan**

# MITOS

MITOS pada SOFTWARE:

- Sikap yang salah
- Menyebabkan masalah yang serius
  - Managers
  - Technical



# MITOS PADA MANAJEMEN



## Mitos:

Kita telah memiliki buku yang dipenuhi dengan berbagai standard dan prosedur untuk membuat perangkat lunak, bukankah hal tersebut sudah cukup untuk diberikan kepada team saya hal-hal yang perlu mereka ketahui

## Fakta:

Apakah buku-buku tersebut memang benar telah digunakan?

Apakah para praktisi perangkat lunak sadar akan keberadaannya?

Apakah lengkap?

Apakah kontennya mencerminkan praktek rekayasa perangkat lunak terkini?

Mayoritas  
Jawabannya adalah  
**TIDAK!**

# MITOS PADA MANAJEMEN



## Mitos:

Team saya memiliki *software development tools* paling canggih dan juga dilengkapi dengan komputer terbaru



## Fakta:

- Untuk mengembangkan software dengan kualitas terbaik dibutuhkan lebih dari tools, workstation atau PC tercanggih.
- Computer-aided software engineering (CASE) lebih penting dibandingkan hardware untuk mencapai produktivitas dan kualitas yang baik.
- Mayoritas praktisi pengembang SW belum menggunakannya secara efektif

# MITOS PADA MANAJEMEN



## Mitos:

Jika proyek kita terlambat, lakukan penambahan programmer dan kita dapat segera mengejar ketertinggalan jadwal proyek kita. → Konsep Mongolian Horde /Gerombolan Mongolia



## Fakta:

- ✓ Software development bukan proses mekanik seperti manufaktur
- ✓ Menambah orang = Proyek semakin telat
- ✓ Penambahan orang dapat dilakukan **JIKA terencana dan terkoordinasi dengan baik**

# MITOS PADA MANAJEMEN



## Mitos:

Jika proyek perangkat lunak ini di outsource ke 3<sup>rd</sup> party, Saya bisa lebih santai dan membiarkan perusahaan tersebut mengembangkannya 😊



## Fakta:

Jika sebuah organisasi tidak memahami bagaimana mengelola dan mengendalikan proyek perangkat lunak secara internal, Maka dibutuhkan usaha yang jauh lebih banyak ketika proyek perangkat lunak tersebut di outsource ke pihak ketiga.



# MITOS PADA PELANGGAN



## Mitos:

*Sebuah pernyataan tujuan /obyektif yang umum sudah cukup bagi developer untuk mulai membuat/menulis program.*

*Hal- Hal yang detil dapat dibahas nanti.*

## Fakta:

- ✓ Definisi yang buruk pada awal proyek = **KEGAGALAN PROYEK**
- ✓ Deskripsi formal dan detil tentang fungsi, perilaku, kinerja antar muka, kendala perancangan, dan validasi kriteria sangat penting
- ✓ Hanya dapat dicapai setelah ada komunikasi yang bagus dan menyeluruh antara pelanggan dan pengembang *software*



# MITOS PADA PELANGGAN

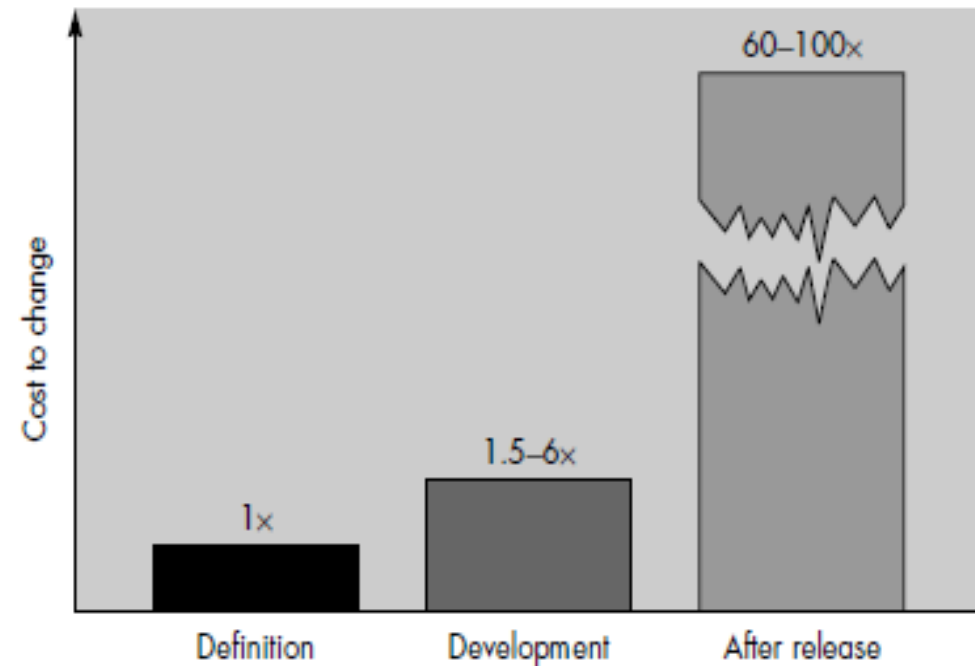


## Mitos:

*Kebutuhan/Persyaratan sebuah proyek berubah secara berkelanjutan, namun perubahan dapat dengan mudah diakomodasi karena perangkat lunak bersifat fleksibel.*

## Fakta:

Memang benar bahwa kebutuhan software berubah, tetapi dampak perubahan bervariasi dengan waktu dimana perubahan tersebut diusulkan.



# MITOS PADA PRAKTIKI

## Mitos:

*Begitu kita menulis /membuat sebuah program dan program tersebut bekerja, tugas kita telah selesai !*

## Fakta:

Semakin cepat anda mulai menulis /melakukan coding program, semakin lama anda akan menyelesaikannya.

60-80% usaha pada perangkat lunak akan dikerahkan ketika dikirimkan pertama kali kepada pelanggan.



# MITOS PADA PRAKTIKI

## Mitos:

*Saya tidak dapat menguji/menilai kualitas dari sebuah program sampai saya mendapati program tersebut 'running'*



## Fakta:

Formal technical review – salah satu mekanisme SQA yang efektif yang dapat diterapkan.

# MITOS PADA PRAKTIKI

## Mitos:

*Satu-satunya indikasi produk yang disampaikan pada customer untuk menandakan proyek tersebut berhasil adalah program yang berjalan dengan baik*



## Fakta:

- ✓ Sebuah “*working program*” hanya salah satu bagian dari konfigurasi perangkat lunak yang meliputi banyak elemen.
- ✓ Dokumentasi memberikan dasar untuk rekayasa SW yang sukses dan, yang lebih penting, pedoman untuk dukungan perangkat lunak.

# MITOS PADA PRAKTIKI

## Mitos:

*Rekayasa perangkat lunak akan membuat kita membuat dokumentasi tebal yang tidak perlu dan selalu akan memperlambat kita*



## Fakta:

- ✓ Software Engineering  $\neq$  Membuat Dokumentasi
- ✓ Menciptakan Kualitas
- ✓ Mengurangi rework
- ✓ Delivery times yang lebih cepat

# SOFTWARE PROCESS

Definisi:

***Software process as framework for the tasks that are required to build high-quality software***

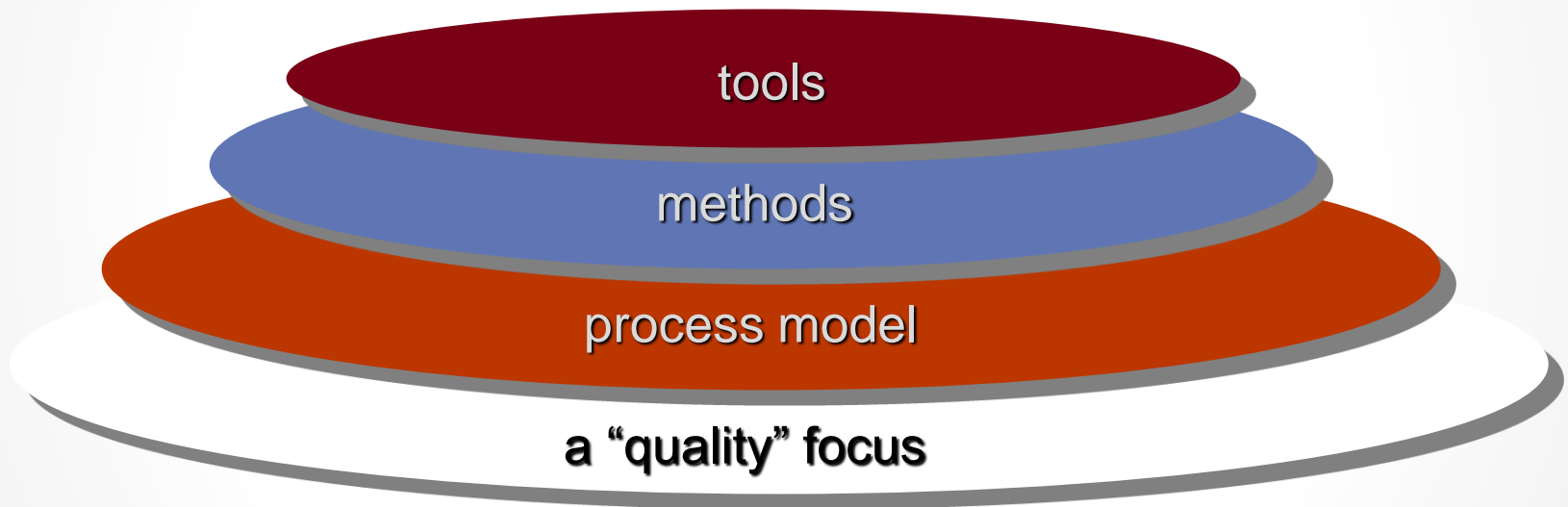
Apakah *software process* ini sama dengan *software engineering*?

## Software Engineering:

- ✓ Kreatif
- ✓ SDM yang memiliki Pengetahuan
- ✓ Dalam proses yang telah mature dan didefinisikan
- ✓ Sesuai dengan produk yang dikembangkan dan tuntutan pasar

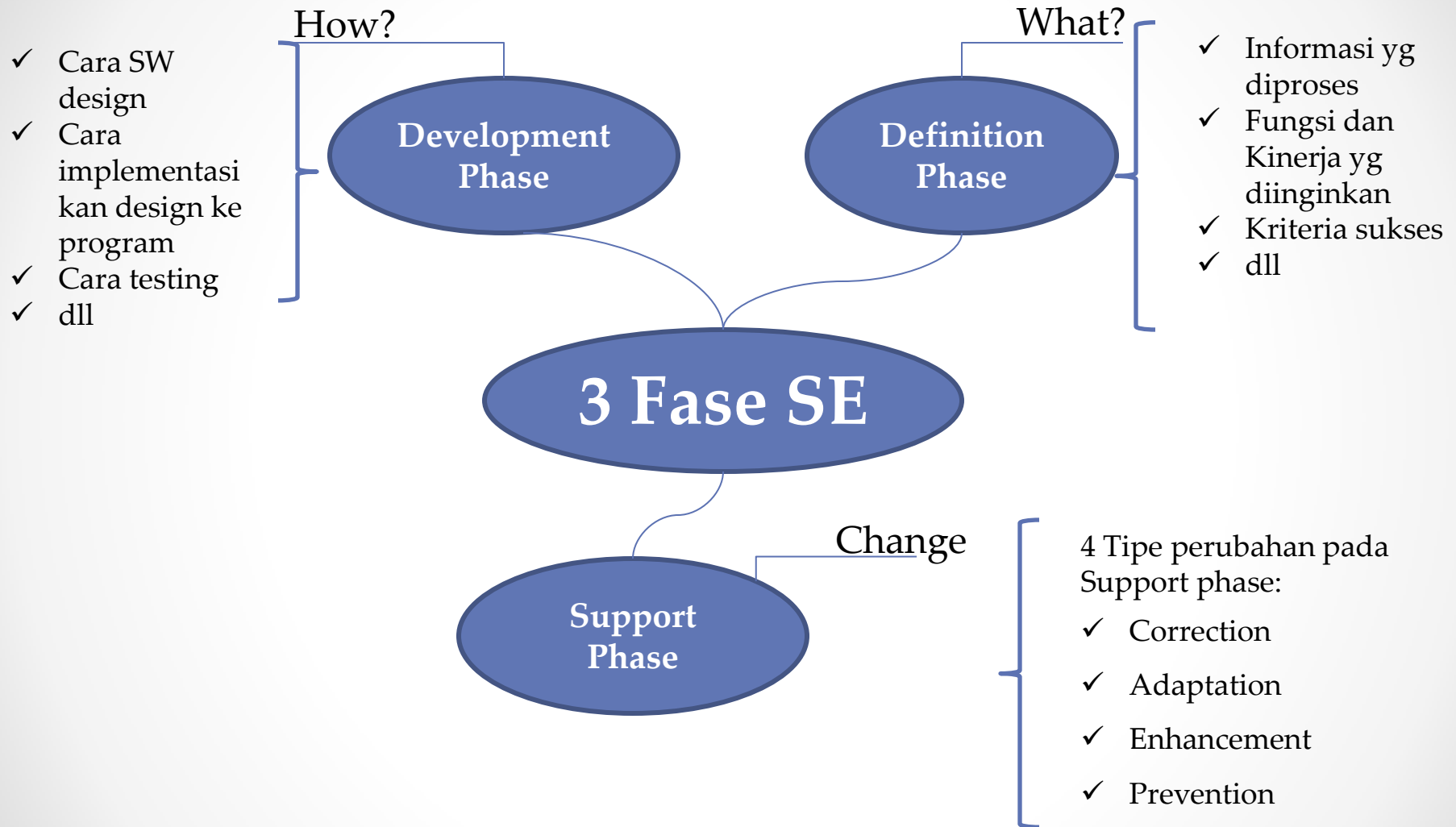
# A Layered Technology

Software Engineering





# GENERIC VIEW OF SE



# Summary

- ✓ Berbagai proses model yang ada masih akan terus berkembang dan berevolusi
- ✓ Model Proses pengembangan SW harus disesuaikan oleh tim pengembang SW.
- ✓ Berbagai tools telah banyak dibuat untuk membantu menggunakan proses model SW ini
- ✓ Jika Proses Lemah = Hasil Deliverable/Produk tidak pasti
- ✓ Namun jika terlalu obsesfi dan kaku dalam menerapkan proses juga berbahaya

# DAFTAR PUSTAKA

- Pressman, RS. Software Engineering: A Practitioner's Approach. Edisi 5. New York. McGraw-Hill. 1992.

# TERIMA KASIH

