

KONSEP MANAJEMEN PROYEK

Perancangan Perangkat Lunak

*Bertalya
Program Pasca Sarjana, Universitas Gunadarma*

Konsep Manajemen Proyek

- Manajemen proyek perangkat lunak merupakan layer pertama pada proses *software engineering* & sangat penting untuk kesuksesan proyek
- 4 P
 1. *People*; manusia sebagai element yang sangat penting untuk kesuksesan proyek
 2. *Product*; software yang akan dibuat
 3. *Process*; serangkaian aktifitas dalam *software engineering*
 4. *Project*; semua pekerjaan yang dibutuhkan untuk mewujudkan produk perangkat lunak

Aspek2 Yang Diperhatikan Manajemen Proyek



Faktor2 Yang Mempengaruhi Hasil Akhir Proyek

- Ukuran
 - deadline penyelesaian
 - anggaran & biaya
 - domain aplikasi
 - teknologi yg diimplementasikan
 - konstrain sistem
 - kebutuhan2 pengguna
 - sumber daya yang tersedia

Aktifitas pada Manajemen Proyek

- Manajemen Proyek terdiri atas
 - *measurement*
 - *estimation*
 - *risk analysis*
 - *scheduling*
 - *tracking & control*
- 1. *Measurement* (Pengukuran)
Pengukuran pada proses dan produk yang dihasilkan
 - biaya

Aktifitas pada Manajemen Proyek

2. *Estimation* (Estimasi)

Di tahap *planning* (perencanaan) pada manajemen proyek, dibuat penaksiran mengenai

- tenaga manusia (jml orang-bulan)
- durasi proyek secara kronologis (kalender)
- biaya

Umumnya, kebanyakan teknik estimasi mempunyai ciri :

- ruang lingkup proyek ditetapkan di awal
- *software metric* (pengukuran lama) sebagai basis estimasi yang baru
- proyek dipecah menjadi bagian kecil yang akan ditaksir secara individual

(Lanjutan)

3. *Risk Analysis*

Rangkaian langkah2 manajemen resiko yang dapat 'menyerang' resiko :

- *risk identification*
- *risk assessement*
- *risk prioritization*
- *risk management strategies*
- *risk resolution & risk monitoring*

(Lanjutan)

4. *Scheduling*

Penjadualan terdiri atas aktifitas

- identifikasi tugas2 proyek
- menentukan saling ketergantungan antara tugas2 proyek
- menetapkan tenaga usaha untuk masing2 tugas
- penunjukkan orang & sumber daya lain
- pembuatan jaringan tugas
- pembuatan penjadualan *time-line*

(Lanjutan)

4. *Tracking & Control*

Manager proyek melakukan penelusuran setiap tugas pada jadwal yang telah ditetapkan

Pengukuran Perangkat Lunak

- Perangkat lunak diukur untuk alasan sbb:
 - mengindikasikan kualitas produk
 - menaksir produktifitas orang2nya
 - menaksir keuntungan (produktifitas & kualitas) dari metode & perangkat2 yang baru
 - memformulasikan baseline utk estimasi
 - membantu justifikasi permohonan perangkat yang baru & training tambahan
- 2 macam pengukuran : *direct measures & indirect measures*

Direct & Indirect Measures

- *Direct measures* pada produk termasuk line of code (LOC), kecepatan eksekusi, ukuran memori, kesalahan pada setiap periode waktu
Contoh : *size-oriented metrics*
 - produktifitas = $\text{KLOC}(\text{thousand line of codes}) / \text{orang-bulan}$
 - kualitas = $\text{defects(kesalahan)} / \text{KLOC}$

Contoh : Size-oriented Metrics

Functions	estimated LOC	LOC/pm	\$/LOC	Cost	Effort (months)
UICF	2340	315	14	32,000	7.4
2DGA	5380	220	20	107,000	24.4
3DGA	6800	220	20	136,000	30.9
DSM	3350	240	18	60,000	13.9
CGDF	4950	200	22	109,000	24.7
PCF	2140	140	28	60,000	15.2
DAM	8400	300	18	151,000	28.0
Totals	33,360			655,000	145.0

Direct & Indirect Measures

- *Indirect measures* pada produk termasuk fungsionalitas, kualitas, kompleksitas, efisiensi, reliability, maintainability

Contoh : *function-oriented metrics*

- produktifitas = $FP(\text{function point}) / \text{orang-bulan}$
- kualitas = $\text{defects(kesalahan)} / FP$

Contoh : Function-oriented Metrics

<u>measurement parameter</u>	<u>count</u>	<u>weight</u>	
number of user inputs	40	x 4	= 160
number of user outputs	25	x 5	= 125
number of user inquiries	12	x 4	= 48
number of files	4	x 7	= 28
number of ext.interfaces	4	x 7	= 28
algorithms	60	x 3	= 180
count-total	→ 569		
complexity multiplier	.84		
feature points	→ 478		

×

0.25 p-m / FP = 120 p-

LOC

Rata2 LOC untuk FP

Assembly	300
Cobol	100
Fortran	100
Pascal	90
Ada	70
Object-oriented	30

Kategori Pengukuran Perangkat Lunak

Kategori I

1. *Productivity metrics*
Pengukuran output dr proses per. lunak
2. *Quality metrics*
Indikasi untuk mengetahui kinerja per. lunak mendekati kebutuhan klien secara implisit & eksplisit
3. *Technical metrics*
Menfokuskan pada karakter per. lunak

Kategori

Pengukuran Perangkat Lunak

Kategori II

1. *Size-oriented metrics*
Mengumpulkan pengukuran output & kualitas *software engineering*
2. *Function-oriented metrics*
Memfokuskan pada fungsionalitas program
3. *Human-oriented metrics*
Mengenai hal2 yg berhubungan dgn manusia dalam *software engineering*

Metric untuk Kualitas Perangkat Lunak

- Penentuan faktor2 kualitas dilihat dr sudut pandang :
 - pengoperasian produk
 - revisi produk
 - transisi produk

Metric untuk Kualitas Perangkat Lunak

- Faktor2 kualitas tsb al:
 1. *Correctness*; program harus beroperasi secara benar
Pengukurannya : kesalahan2 program per KLOC (*thousand lines of code*) dalam satu periode
 2. *Maintainability*; kemudahan dalam memperbaiki kesalahan program, beradaptasi dengan perubahan lingkungan, berkembang sesuai dengan keinginan klien

(lanjutan)

Pengukurannya dg *indirect measure*, berorientasi pada waktu; yakni MTTC (*mean time to change*), waktu analisis utk perubahan, perancangan sesuai modifikasi, implementasi, testing & distribusi

3. *Integrity*, keintegritasan per. lunak dibutuhkan utk melawan serangan dari hacker & virus.

Pengukurannya :

$$\sum [1 - \text{threat} \times (1 - \text{security})]$$

- *threat* : probabilitas serangan sesuatu pada waktu tertentu
- *security* : probabilitas serangan berulang lagi

(lanjutan)

4. *Usability*, perangkat lunak '*user friendly*'.

Pengukurannya dg 4 karakteristik :

- kemampuan intelektual & fisik yg dibutuhkan utk mempelajari sistem
- waktu yg dibutuhkan utk efisiensi penggunaan sistem
- produktifitas bersih pada saat sistem digunakan oleh seseorang yg efisien
- penaksiran subjektif terhadap tingkah laku pengguna pada sistem

5 Faktor yang Mempengaruhi Produktifitas

1. Orang-orang nya; ukuran dan kepakaran dari organisasi pembuatan perangkat lunak
2. Permasalahan; kompleksitas permasalahan & banyaknya perubahan pada kebutuhan perancangan
3. Proses; teknik analisis & perancangan yg digunakan, bahasa & perangkat CASE yg ada
4. Produk; reliabilitas dan kinerja dari sistem berbasis komputer
5. Sumber daya; ketersediaan perangkat CASE, sumber daya perangkat lunak & keras

Penetapan Baseline

- *Baseline* merupakan sebuah tabel sederhana atau template yang terdiri atas data yg dikumpulkan dari proyek perangkat lunak lama.
- Untuk membantu perencanaan strategis & estimasi biaya, data baseline harus bersifat
 - data harus akurat
 - data dikumpulkan sebanyak2nya proyek
 - pengukuran harus konsisten
 - aplikasi yang digunakan utk bekerja harus sama dengan estimasi

3 langkah

Memulai program Metric

1. Pengumpulan data; yang membutuhkan investigasi sejarah proyek yang lama untuk merekonstruksi data yg dibutuhkan
2. Perhitungan metric; dengan pengukuran LOC atau FP
3. Evaluasi data; mengevaluasi dari data yang sudah dihitung

selesai

Referensi

1. Pressman Pressman, Roger S , Roger S, ,
Software Engineering : Software Engineering : A Practitioner A Practitioner's Approach s Approach, McGraw Hill, , McGraw Hill, 2002
2. Sommerville Sommerville, Ian , Ian, ,
Software Engineering Software Engineering, , Addison Wesley, 1992