

Nama : Rendy Galih Saputra
Kelas : 5C
NPM : 1710631170176

- Judul survey paper : *Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey*
- Penulis : Faiza Anwer, Shabib Aftab, Usman Waheed dan Syed Shah Muhammad
- Tahun : 2017
- Penerbit : Department of Computer Science, Virtual University of Pakistan
- Alamat web download:
https://www.researchgate.net/profile/Shabib_Aftab/publication/316273992_Agile_Software_Development_Models_TDD_FDD_DSDM_and_Crystal_Methods_A_Survey/links/58f86bc44585158d8a6c4f11/Agile-Software-Development-Models-TDD-FDD-DSDM-and-Crystal-Methods-A-Survey.pdf

Ringkasan Survei Paper

Abstrak

Model proses pengembangan perangkat lunak yang baik dapat memainkan peran yang sangat penting dalam mengembangkan perangkat lunak yang memiliki kualitas tinggi. Dalam *paper* ini memberikan tinjauan dari beberapa *agile methodology* atau metodologi tangkas yang biasa digunakan dalam industri perangkat lunak.

I. Pengenalan

Metodologi pengembangan perangkat lunak tangkas/gesit bertujuan untuk mengembangkan perangkat lunak yang dapat mengatasi keterbatasan model pengembangan perangkat lunak tradisional. Istilah gesit/tangkas ini diciptakan tahun 2001 ketika 17 pengembang perangkat lunak terkenal bertemu di Utah untuk mengeksplorasi cara-cara baru dan peningkatan pengembangan perangkat lunak. Metodologi pengembangan perangkat lunak tangkas/gesit ini lebih memfokuskan terhadap orang-orang dan menimbang hal-hal yang diabaikan dalam model tradisional.

Pernyataan sikap gesit/tangkas didefinisikan menjadi 12 prinsip yang menjadi fondasi dari pengembangan perangkat lunak gesit. 12 prinsip ini adalah:

1. Kepuasan pelanggan saat diawal dan berkelanjutan.
2. Menyambut perubahan bahkan diterapkan dalam tahap pengembangan selanjutnya.
3. Bekerja sama dan berkomunikasi antara pelanggan dan pengembang.
4. Pengiriman software yang berfungsi secara rutin.
5. Menyemangati dan memotivasi orang yang bekerja dalam pengembangan software.
6. Komunikasi secara langsung.
7. Tujuan utama adalah perangkat lunak yang bekerja.

8. Mempertahankan kecepatan melalui pengembangan yang berkelanjutan.
9. Memperhatikan desain yang baik secara terus-menerus.
10. Menjaga agar hal-hal tetap sederhana.
11. Tim yang dapat mengorganisir sendiri dapat memunculkan arsitektur, kebutuhan, dan rancangan perangkat lunak yang terbaik.
12. Sebuah tim secara berkala berdiskusi untuk menemukan sebuah cara efektif.

II. Proses Model

a) *Test Driven Development*

Membangun perangkat lunak dengan cara melakukan tes dan perbaikan secara terus-menerus hingga perangkat lunak tersebut sesuai dengan permintaan.

- Siklus dalam TDD

1. **Menuliskan tes** : Dalam TDD, siklus pertama dimulai saat *developer* menuliskan tes yang akan dilakukan. Hal ini, dapat membantu *developer* agar dapat berfokus pada kebutuhan yang akan diimplementasikan pada program nantinya.
2. **Menjalankan semua tes untuk mendeteksi kegagalan** : *Developer* mulai menjalankan tes yang tadi telah dituliskan dan mendeteksi kegagalan dalam menjalankan tes tersebut.
3. **Tulis kode**: Jika sudah menemukan kegagalan, *developer* menulis ulang kode tersebut sampai berhasil menyelesaikan kegagalan. Jika semua tes berhasil, maka akan berlanjut ke tahap selanjutnya. Jika gagal, *developer* harus menulis kode lagi sampai berhasil memecahkan masalahnya.
4. **Refactor kode**: Setelah berhasil menyelesaikan semua masalah, langkah terakhir adalah menyederhanakan kode yang telah dibuat tadi.

- Kelebihan TDD

- Uji pertama dalam TDD membantu menemukan kecacatan sebelumnya dan dekat dengan asalnya. Sehingga, kualitas kode dapat ditingkatkan.
- TDD meningkatkan efisiensi karena saat bertemu dengan masalah, TDD akan langsung melakukan perbaikan.
- Mengurangi kompleksitas kode. Karena, melakukan penyederhanaan ketika sudah selesai melakukan semua proses.
- Perbaikan langsung ketika melakukan kesalahan dapat memberikan kesempatan programmer untuk lebih berkonsentrasi dan mengembangkan lebih efektif.

- Kekurangan TDD

- Programmer harus memiliki keahlian khusus untuk melakukan TDD.
- TDD tidak dapat digunakan dalam segala situasi.
- TDD akan memakan banyak waktu jika terjadi kesalahan secara terus-menerus.

b) *Feature Driven Development*

FDD adalah jenis membangun perangkat lunak yang berfokus pada desain dan fase pembangunan.

- Siklus FDD
 1. **Membuat keseluruhan model** : Dalam langkah ini, semua anggota tim mencoba untuk membuat keseluruhan model yang akan dibuat. Hal ini berguna agar anggota tim dapat berfokus tentang program yang akan dibuat nantinya.
 2. **Buat daftar fitur** : Setelah membuat keseluruhan model, langkah selanjutnya adalah membuat daftar fitur yang akan dibuat dalam proyek tersebut.
 3. **Membuat rencana berdasarkan fitur** : Pada bagian ini, manajer proyek, manajer pengembangan, dan ketua *programmer* mendiskusikan tentang rencana fitur yang akan dibuat terlebih dahulu. Hal ini diperlukan agar nantinya proyek tersebut dapat selesai sesuai jadwal.
 4. **Desain fitur** : Setelah membuat rencana dari fitur yang akan dibuat, maka langkah selanjutnya adalah membuat desain dari fitur tersebut. Setelah desain fitur tersebut mendapat persetujuan, maka proyek dapat melangkah ke tahap selanjutnya.
 5. **Membuat fitur** : Dalam tahap ini, semua aspek ikut dilibatkan. Selain itu, kegiatan pengujian kode dan pengujian produk dilakukan pada tahap ini.
- Kelebihan FDD
 - Tim FDD berfokus dalam kualitas melalui fase pengembangan.
- Kekurangan FDD
 - FDD memerlukan tim dengan kemampuan tinggi dan memiliki kelebihan dalam desain dan memodelkan sesuatu.

c) *Dynamic System Development Model*

DSDM adalah sebuah metode pengembangan sistem yang berfokus pada peningkatan kualitas. Sama seperti metode sebelumnya, metode ini melakukan proses secara berulang-ulang untuk memberikan perangkat lunak berkualitas.

- Siklus DSDM
 1. **Pra proyek** : Pada tahap ini, proyek di pilih berdasarkan ketepatan DSDM. Selain itu juga, pada tahap ini mendiskusikan tentang perkiraan waktu dan biaya yang akan digunakan.
 2. **Studi kelayakan** : Pada tahap ini, proyek tersebut akan didiskusikan uji kelayakannya. Mulai dari resiko yang ditimbulkan, perusahaan yang akan menggunakannya, sampai kelayakan anggotanya.
 3. **Studi bisnis** : Pada tahap ini, akan diadakan diskusi dari sisi *user* mulai dari model proyeknya, struktur sistemnya, sampai pada proyek tersebut dapat dikembangkan di masa depan nantinya.

4. **Pemodelan fungsional** : Tahap ini adalah tahap berulang, dimana pada tahap ini proses koding dilakukan. Selain itu tahap ini juga dilakukan pemodelan dari aktivitas yang dapat dilakukan nantinya.
5. **Desain dan buat** : Tahap ini adalah tahap berulang yang perulangannya tergantung di tahap sebelumnya. *User* akan menguji dan menganalisa sistem yang telah dibuat pada tahap sebelumnya. Jika ada kesalahan maka akan diperbaiki sampai sesuai dengan ketentuan yang telah ditentukan.
6. **Implementasi** : Pada tahap ini, hasil sistem yang sudah jadi tadi, diserahkan kepada *user*. Selain itu, pihak *developer* juga menyerahkan buku manual dan panduan tentang sistem tersebut.
7. **Pasca proyek** : Setelah proyek tersebut resmi dibubarkan, penilaian terhadap sistem yang di buat diperlukan sebagai acuan sejauh mana tingkat keberhasilan sistem tersebut terhadap keuntungan perusahaan.

- Kelebihan DSDM
 - DSDM menyediakan pengembangan aplikasi yang cepat berdasarkan prinsip-prinsip gesit/tangkas.
 - Menyediakan pedoman yang lebih baik untuk aspek proyek lainnya.
- Kekurangan DSDM
 - Sejumlah besar peran dalam DSDM dapat menciptakan masalah administrasi selama proses pengembangan.
 - Sebagai kerangka kerja, DSDM tidak menyediakan pedoman yang spesifik tentang isu yang berkaitan dengan besar tim dan panjang perulangan.

d) *The Crystal Methods*

Metode kristal adalah salah satu metode pengembangan perangkat lunak gesit/tangkas yang dapat digunakan untuk proyek-proyek perangkat lunak yang berbeda, tergantung dari ukuran, kompleksitas, kekritisannya, dan banyaknya orang yang terlibat.

- Kelebihan metode kristal
 - Komunikasi tim yang efektif adalah kunci dari keberhasilan suatu proyek. Metode kristal menyediakan pedoman yang tepat tentang komunikasi dari ukuran tim yang bervariasi.
 - Metode kristal dapat digunakan untuk proyek dengan ukuran yang berbeda.
 - Metode kristal memiliki kontrol risiko yang baik.
- Kekurangan metode kristal
 - Metode kristal kekurangan desain dan aktivitas verifikasi kode.
 - Metode kristal tidak menyediakan pedoman tentang bisnis.