

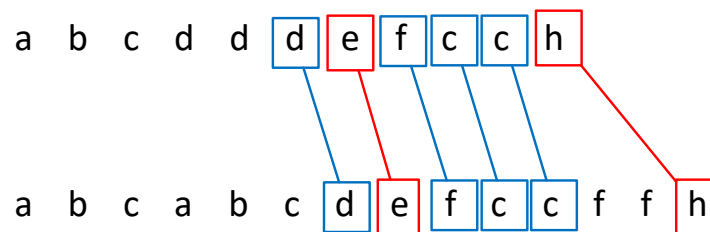
Software Engineering I

Lösungsvorschlag zu Übungsblatt 12

Aufgabe 12.1

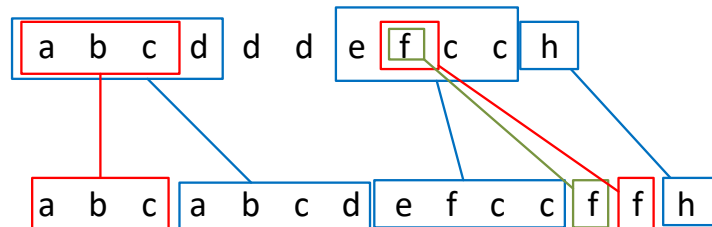
(b) Algorithmus von Heckel

- Identifikation von **eindeutigen** Zeichen in beiden Sequenzen
- **Erweiterung** nach links und rechts



(c) Algorithmus von Tichy

- **Durchlaufen** von s_2
- Suche nach **korrespondierenden Sequenzen** in s_1
- Identifikation **maximaler Blöcke**



Aufgabe 12.2 (Drei-Wege-Verschmelzen)

(a) Operationssequenzen

$\Delta(b, a_1)$:

- INSERT 1 „import j. . . “
- CHANGE 3 „public D. . . “
- CHANGE 4 „public P. . . “

$\Delta(b, a_2)$:

- INSERT 3 „public char g. . . “
- CHANGE 4 „public P. . . “
- INSERT 7 „this.g. . . “
- CHANGE 9 „return n. . . “

(b) **Konflikt:** Inhalt von Zeile 4 wird unterschiedlich geändert

$\Delta(b, a_1)$:

- INSERT 1 „import j...“
- CHANGE 3 „public D...“
- **CHANGE 4 „public P...“**

$\Delta(b, a_2)$:

- INSERT 3 „public char g...“
- **CHANGE 4 „public P...“**
- INSERT 7 „this.g...“
- CHANGE 9 „return n...“

(c) **Verschmolzene Version**

$\Delta(b, m)$:

- | | |
|-------------------------------|---------------------------------|
| • INSERT 1 „import j...“ | • CHANGE 4 „public P...“ |
| • INSERT 3 „public char g...“ | • INSERT 7 „this.g...“ |
| • CHANGE 3 „public D...“ | • CHANGE 9 „return n...“ |

m :

```
import java.util.Date;
public class Person {
    public String name;
    public char geschlecht;
    public Date geburtsdatum;
    public Person(String name, Date geburtsdatum) {
        this.name = name;
        this.geburtsdatum = geburtsdatum;
        this.geschlecht = geschlecht;
    }
    public String toString() {
        return name + "┘" + geburtsdatum.toString()
            + "┘" + geschlecht;4
    }
}
```

Aufgabe 12.3 (Softwareprozesse)

- (a) Nennen Sie die **sechs Merkmale** zur Klassifikation von Entwicklungsprozessen. Nennen Sie **jeweils eine Ausprägung** für jedes Merkmal.

| Merkmal | Ausprägung (eine genügt) |
|--------------------|---|
| Lebenszyklusmodell | Wasserfall, iterativ, inkrementell, Arbeitsbereiche |
| Planung | agil, plangetrieben |
| Granularität | fein, mittel, grob |
| Arbeitsbereiche | Requirements Engineering, Entwurf, Implementierung |
| Strukturiertheit | schwach strukturiert, stark strukturiert |
| Größe | klein, mittel, groß |

- (b) Klassifizieren Sie die Entwicklungsprozesse **XP** und **RUP** nach den von Ihnen genannten Merkmalen.

| Merkmal | XP | RUP |
|--------------------|--------------------------------------|----------------------|
| Lebenszyklusmodell | iterativ, inkrementell, keine Phasen | iterativ, mit Phasen |
| Planung | agil | plangetrieben |
| Granularität | mittel | fein |
| Arbeitsbereiche | alle | alle |
| Strukturiertheit | schwach | stark |
| Größe | klein | alle Teamgrößen |

- (c) Nennen und beschreiben Sie kurz die Rollen in Scrum.

- **Product Owner** - definiert Anforderungen, verhandelt mit dem Team über die Sprint Ziele.
- **Scrum Master** - unterstützt den PO in der Projektplanung, unterstützt das Development Team bei selbstorganisierender Arbeit. Zuständig für das Einhalten der Scrum Regeln.
- **Development Team** - 3-9 Personen, selbstorganisierende Produktentwicklung, Planung des Sprints.

- (d) Nennen und beschreiben Sie kurz vier Scrum Artefakte. Nennen Sie die verantwortliche Rolle für jedes Artefakt.

- **Product Backlog** - priorisierte Liste aller Anforderungen an das Produkt. Product Owner.
- **Sprint Backlog** - Liste der in einem Sprint zu realisierenden Anforderungen. Development Team.
- **Produktinkrement** - Im Rahmen eines Sprints realisierte Produktfunktionalität. Development Team.
- **Definition von abgeschlossen** - Festlegung der Kriterien, die erfüllt sein müssen, um potenziell auslieferbare Produktfunktionalität zur Verfügung zu stellen. Alle.

- (e) Beschreiben Sie kurz, was ein Sprint ist. Wie lange kann ein Sprint maximal dauern?

Ein Sprint ist eine Iteration des Prozesses, in der ein potenziell auslieferbares Inkrement der Produktfunktionalität entwickelt wird. Maximale Dauer: 30 Tage.

Aufgabe 12.4 (Zusatzaufgabe:LCS Algorithmus)

Erstellen Sie ein UML-Aktivitätsdiagramm, das den Algorithmus zur Berechnung der LCS-Matrix Einträge auf der Ebene der elementaren Java Anweisungen beschreibt. Der Algorithmus befindet sich im elearning bei Aufgabe 11. Verzichten Sie auf Objektflüsse und Partitionen. Gehen Sie von folgendem strukturellen Modell aus:

