

BIG B4NG challenge, 17. Wettbewerb Aufgabe 1

Diese Aufgabe wird vom Fachgebiet Software Engineering an der Fakultät für Elektrotechnik und Informatik der Leibniz Universität Hannover gestellt.

Weitere Informationen zum Fachgebiet Software Engineering findet ihr unter <http://www.se.uni-hannover.de>

Mit Graphen zu einem erfolgreichen Projektabschluss

Ein Graph ist eine mathematische Struktur, die aus einer Menge von Objekten besteht. Diese Objekte stehen in Verbindung miteinander. Wenn man einen Graphen visualisiert, werden die Objekte durch Knoten dargestellt, die durch Kanten miteinander verbunden sind. Ein Beispiel dafür seht ihr im rechten Bild.

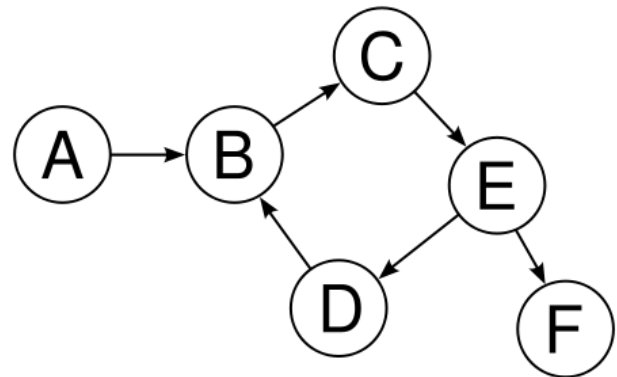


Abbildung 1: Ein gerichteter Graph
Quelle: Wikipedia (Graphentheorie)

Graphen spielen nicht nur in der Mathematik eine besondere Rolle. Auch Fahrpläne, Stammbäume und die Struktur von Molekülen können als Graphen aufgefasst werden. In der Informatik tauchen Graphen an verschiedensten Stellen auf: Bei Ablaufdiagrammen von Programmen, Netzplänen und Netzwerken von Entwicklern handelt es sich beispielsweise um Graphen, die oft mit gängigen Methoden der Graphentheorie analysiert werden können.

In dieser Aufgabe beschäftigen wir uns damit, wo Graphen im Software Engineering Anwendung finden, um bspw. die Planung eines Softwareprojekts zu erleichtern oder einen Überblick über Informationsflüsse zu gewinnen.

a) Die Planung vor einem Projekt (10 Punkte)

Bevor ein Projekt starten kann, müssen viele Dinge geklärt werden. Dazu gehört bspw. die Anzahl und Auswahl an Mitarbeiterinnen und Mitarbeitern, die an dem Projekt beteiligt sind, das zur Verfügung stehende Budget und der Zeitplan.

Meistens hängt die Planung stark von der gewählten Arbeitsorganisation ab: Entwickelt man plangetrieben, agil oder hybrid?

- 1) Beschreibt in wenigen Sätzen, was die Begriffe „agile“, „hybride“ und „plangetriebene“ Software-Entwicklung bedeuten.
- 2) Nennt jeweils einen Vor- und Nachteil für die zuvor genannten Arten von Arbeitsorganisation.

Wenn die Arbeitsorganisation gewählt und die organisatorischen und vertraglichen Details geklärt sind, kann man dazu übergehen, die zu erfüllenden Aufgaben in Arbeitspaketen zusammenzufassen. Diese hängen jedoch sehr oft voneinander ab. Das heißt, dass bspw. AP2 erst beginnen kann, wenn AP1 abgeschlossen ist. Oft legt man für den Abschluss einzelner Arbeitspakete eine Deadline, d. h. einen festen Termin für die Abgabe oder Fertigstellung, fest und verknüpft diese mit einem Meilenstein, um den Überblick über den Fortschritt des Projekts zu behalten: Wird ein Meilenstein nicht rechtzeitig erreicht, ist das Projekt in Verzug und man sollte gegensteuern.

Die Abhängigkeiten von den Arbeitspaketen werden oft als Netzplan visualisiert. Dabei werden die einzelnen Arbeitspakete als Kanten dargestellt. Einen beispielhaften Netzplan sieht ihr in Abbildung 2.

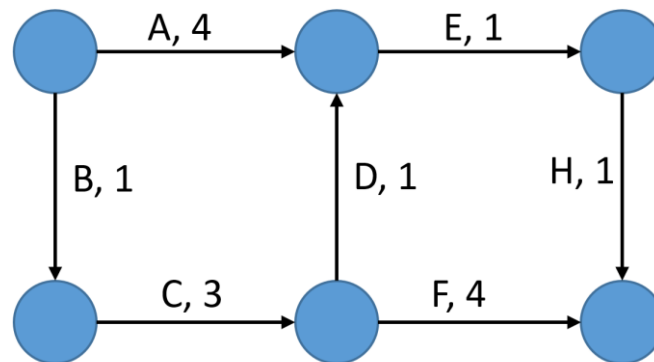


Abbildung 2: Netzplan

3) Zeichnet den Netzplan, der zu folgendem Projekt passt:

Im Rahmen des Projekts soll eine digitale Bibliothek implementiert werden. Dazu ist es zunächst notwendig, die Anforderungen mit dem Kunden zu klären (a). Dieser Schritt dauert 3 Wochen. Danach wird die Entwicklungsumgebung (b) aufgesetzt. Parallel dazu finden die so genannten Spikes (c) statt. Während der Spikes machen sich die Entwickler mit neuen Technologien (wie bspw. einem Scanner) vertraut. Dafür ist eine Woche angesetzt. Für die Entwicklungsumgebung sind zwei Wochen geplant. Sobald die Entwicklungsumgebung steht, wird analog die Testumgebung aufgesetzt (d). Das dauert eine Woche. Nach den Spikes und dem Aufsetzen der Entwicklungsumgebung können die Entwickler mit der Implementierung beginnen (e). Die Implementierung dauert 5 Wochen. Danach wird getestet (f). Dafür muss die Testumgebung fertig sein. Das Testen dauert 2 Wochen. Nach dem Testen geht die Software für zwei Wochen in den Praxiseinsatz (g). Danach werden Bugs, d. h. Fehler im Programm, gefixt (h). Das dauert eine Woche. Im Anschluss wird das finale Produkt an den Kunden ausgeliefert.

Die Netzpläne eignen sich sehr gut, um die kürzeste Dauer des Projekts, den Puffer bei einzelnen Arbeitspaketen und den so genannten *kritischen Pfad* zu berechnen. Ein *Pfad* ist definiert als eine Folge von Knoten, in der jeweils zwei aufeinanderfolgende Knoten durch eine Kante miteinander verbunden sind. Dabei müssen nicht alle Knoten abgelaufen werden. Der kritische Pfad ist der Pfad aller Aktivitäten, die keinen Puffer haben. Eine Verzögerung auf diesem Pfad führt also zu Verzug.

- 4) Berechnet für euren Netzplan die kürzeste Dauer des Projekts, für jeden Knoten den Puffer und kennzeichnet den kritischen Pfad. Erklärt kurz, wie ihr auf eure Lösung gekommen seid.
- 5) Wie viele Mitarbeiter braucht man mindestens, um das Projekt frühestmöglich fertig zu stellen?

Hinweis: Solltet ihr bei Aufgabenteil 3 auf keine Lösung gekommen sein, könnt ihr bei den Aufgaben 4 und 5 den Netzplan aus Abbildung 2 verwenden.

b) Zusammenarbeit während eines Projekts (12 Punkte)

Vor allem in größeren Projekten entwickelt meist ein Team von Entwicklern die gewünschte Software.

- 1) Nennt drei Gründe, warum Software-Entwicklung Teamarbeit erfordert.
- 2) Gibt es Situationen, in denen es besser ist, wenn ein einzelner Entwickler an dem Projekt arbeitet? Nennt mindestens ein Beispiel.

In Teams ist es wichtig, dass jedes Teammitglied alle Informationen enthält, die es benötigt. Das können zum Beispiel geänderte Kundenanforderungen, Richtlinien oder organisatorische Informationen sein.

Um den Informationsfluss in einem Team während des Projekts zu visualisieren, wurde am Fachgebiet Software Engineering die *FLOW-Methode* entwickelt. Sie hilft dabei, Informationsflüsse zu verstehen, zu visualisieren, zu analysieren und im Anschluss ggf. zu verbessern. FLOW unterscheidet zwischen sogenannten *festen und flüssigen Informationsspeichern*. In den Informationsspeichern sind Informationen hinterlegt. Dabei sind feste Informationen wiederholt und langfristig abrufbar und auch für Dritte verständlich. Bei flüssigen Informationen ist mindestens eine der zuvor genannten Eigenschaften verletzt.

- 3) Gebt je drei Beispiele für feste und flüssige Informationsspeicher an. Begründet eure Entscheidung. Wie verhält es sich mit E-Mails?

In Interviews mit drei bis sieben geeigneten Personen (Projektleiter, Teammitglieder mit besonderen Aufgaben, „normale“ Teammitglieder) wird geklärt, auf welchen Wegen Informationen innerhalb des Projektteams weitergegeben werden. Dabei können die Interviews sowohl vor, während als auch im Anschluss an das Projekt geführt werden. Zu Beginn kann der Sollzustand ermittelt werden; zu den beiden anderen Zeitpunkten der Ist-Zustand.

Informationen zu FLOW und zur Notation findet ihr unter

http://www.se.uni-hannover.de/pages/de/projekte_flow.

Ein Beispiel für ein FLOW-Diagramm findet ihr in Abbildung 3. Die Knoten im Diagramm sind die Personen und Dokumente, die an dem Projekt beteiligt sind, und die Kanten stehen für eingehende und ausgehende Informationsflüsse. Ein FLOW-Analyst betrachtet nun die einzelnen Knoten im Diagramm, um kritische Punkte zu identifizieren.

- 4) Knoten mit besonders vielen ein- oder ausgehenden Kanten sind wichtig für das Projekt, stellen aber u. U. auch ein erhöhtes Risiko dar. Erklärt in wenigen Sätzen, warum das so ist.

Zur Auswertung der Diagramme kann auf so genannte *Zentralitätsmaße* aus der sozialen Netzwerkanalyse zurückgegriffen werden. Zentralitätsmaße dienen der Analyse von Strukturen in sozialen Netzwerken. Sie können aber auf Rechnungen auf Graphen und damit auf die Graphentheorie zurückgeführt werden. Im Gegensatz zur Graphentheorie, die oftmals theoretische Anwendungen hat, steht bei den Zentralitätsmaßen jedoch die Interpretation im Vordergrund. Deshalb ist die Berechnung oft nicht trivial.

- 5) Recherchiert drei Zentralitätsmaße und erklärt kurz, was sie angeben. Ihr könnt auch eine Berechnungsformel angeben. Wichtiger ist jedoch, was berechnet wird.
- 6) Betrachtet die Berater in dem Projekt (Abbildung 3). Welche der zuvor genannten Zentralitätsmaße würdet ihr verwenden, um ihre zentrale und wichtige Position im Netzwerk quantitativ zu belegen? Gebt beispielhaft In- und Outdegree an.

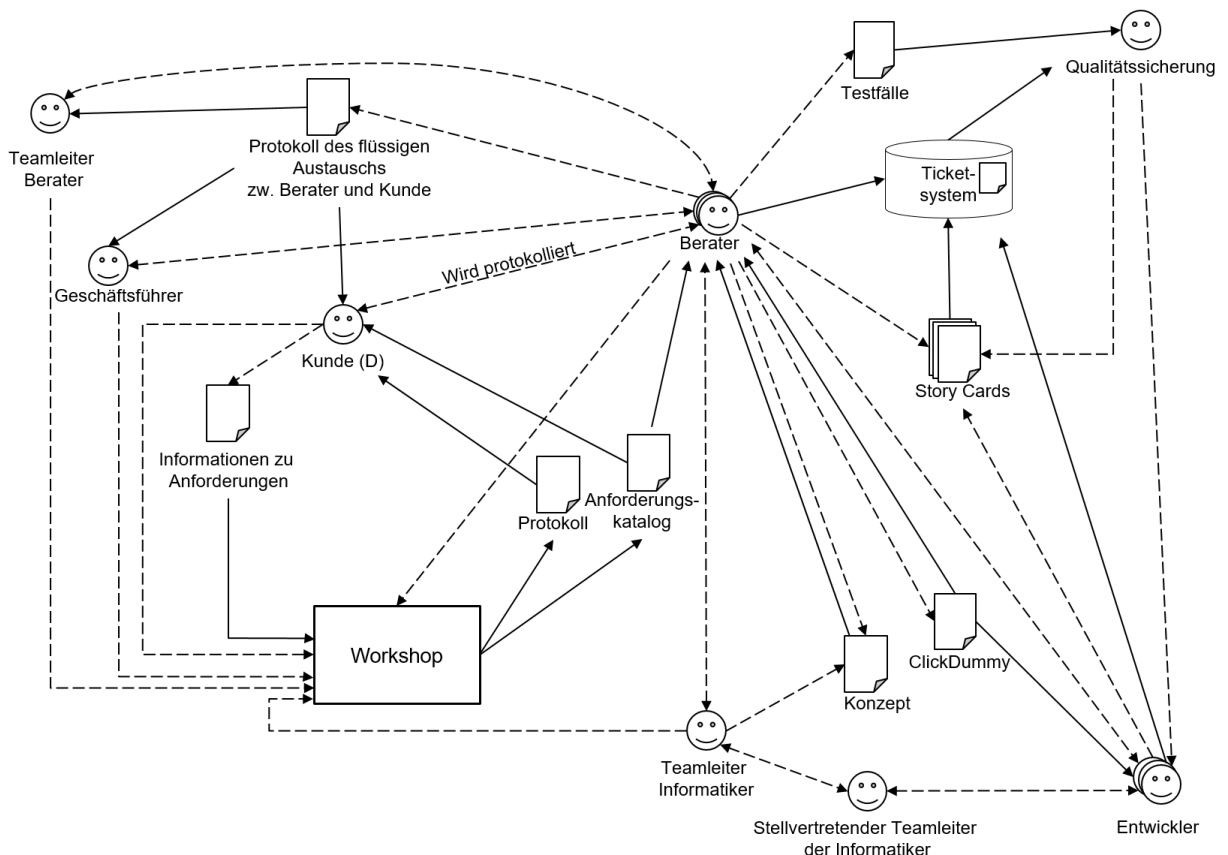


Abbildung 3: FLOW-Diagramm [1]

- 7) Was ist euer Eindruck von dem Informationsfluss in dem dargestellten Projekt? Begründet eure Beobachtungen möglichst mathematisch.
- 8) Würdet ihr dem Projektleiter Änderungen oder Anpassungen empfehlen? Wenn ja, welche?

c) Analyse während des Projekts (8 Punkte)

Man kann ein FLOW-Diagramm, wie ihr es in der vorigen Teilaufgabe kennen gelernt habt, auch nutzen, um den *maximal möglichen Informationsfluss* zu ermitteln. Wie das funktionieren könnte, wurde gerade in einer Bachelorarbeit am Fachgebiet Software Engineering [2] vorgestellt. Dabei wird auf den Algorithmus für den maximalen Fluss von Ford-Fulkerson zurückgegriffen.

Wir betrachten das folgende Netzwerk in Abbildung 4, das aus einem FLOW-Diagramm entstanden ist:

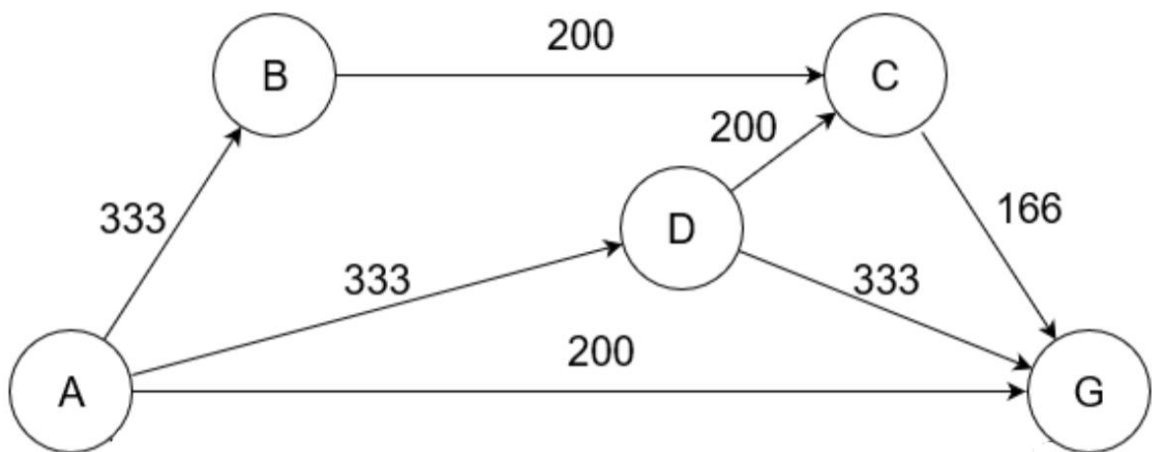


Abbildung 1: Netzwerk aus FLOW-Diagramm [2]

- 1) Wie ist der maximale Fluss in einem Netzwerk definiert?
- 2) Wie kann man den maximalen Fluss berechnen? Überlegt euch oder recherchiert eine Möglichkeit und beschreibt diese ausführlich. Ihr könnt diese entweder in natürlicher Sprache beschreiben, oder ihr nutzt ein Nassi-Shneiderman-Diagramm oder Pseudocode.
- 3) Berechnet den maximalen Fluss vom Knoten A zum Knoten G.

Ein verwandtes Problem zum Problem des maximalen Flusses ist das Problem des minimalen Schnittes.

- 4) Wie ist ein so genannter *minimaler Schnitt* eines Graphen definiert?
- 5) Kennzeichnet einen minimalen Schnitt in dem Graphen aus Abbildung 4.
- 6) Was besagt das max-flow-min-cut-Theorem? Wie kann dieses Theorem bei der Informationsflussanalyse weiterhelfen?

Viel Erfolg!

Referenzen

- [1] Jil Klünder und Kurt Schneider, *Informationsfluss in verteilten Softwareprojekten - Eine Einzelfallstudie*, In PERSONALquarterly, 69(2), 2017.
 - [2] Jonas Wallat: *Vergleich von Algorithmen zur Informationsflussanalyse in der Software-Entwicklung*, Bachelorarbeit, Leibniz Universität Hannover, Fachgebiet Software Engineering, 2017.
-

Allgemeine Hinweise

Einsendeschluss: Sonntag, 22. Oktober 2017, 19:59 Uhr.

Gebt eure Lösungen über das Portal von uniKIK ab: <https://portal.studienberatung.uni-hannover.de/>

Zulässige Dateiformate sind: PDF für die zusammengeschriebene Lösung (mit eingebetteten Bildern), sowie unter Windows gängige Videoformate, die sich ohne Installation von zusätzlicher Software abspielen lassen, z. B. mp4.

Die Dateien sollten nicht größer als 7,5 MB sein (die Dateien können gezippt sein)! Bitte gebt auch euren Teamnamen, die Namen der Gruppenmitglieder sowie deren Schulen an. Bitte benennt eure hochgeladenen Dateien nach dem Gruppennamen.

ACHTUNG bei Zip-Dateien! Um sicher zu gehen, dass eure Dateien wirklich fehlerfrei und für die Korrektoren/-innen zu öffnen sind, solltet ihr eure Zip-Dateien etc. noch mal von eurem Account herunterladen und öffnen. Dateien, die sich nicht öffnen lassen, können nicht bewertet werden!

Gebt eure Lösungen auch dann ab, wenn ihr nicht alle Fragen beantworten konntet! Vielleicht gelingt euch das ja bei den kommenden Aufgaben.

Die Teilnahmebedingungen und weitere Informationen findet ihr unter: <https://www.studienberatung.uni-hannover.de/bigbangchallenge.html>

Der Rechtsweg ist ausgeschlossen.