


# Spielanleitung Tech-Schulden 2.0

## 1 Einleitung

### 1.1 Ziel des Spiels

Im Laufe des Spiels Tech-Schulden 2.0 werden von den Spielenden Softwaresysteme gebaut. Ziel des Spiels ist es, die meisten Punkte für das eigene System zu erlangen. Das Aufnehmen von Technischen Schulden kann einem das Vorhaben erschweren und erleichtern. Während des Spielens soll ein Bewusstsein für die Thematik der Technischen Schulden geschaffen werden. In den gelb hinterlegten Textboxen werden **Bezüge zur Realität** aufgezeigt, gekennzeichnet mit dem Symbol .

Das Spiel wird 2 gegen 2 gespielt, um Diskussionen zwischen den Teammitgliedern anzuregen.

**Wer am besten mit Technischen Schulden umgeht, gewinnt!**



#### Technische Schulden

“In software-intensive systems, technical debt is a collection of design or implementation constructs that are expedient in the short term, but set up a technical context that can make future changes more costly or impossible.”

Technische Schulden sind Konstruktionen in der Softwareentwicklung, die zunächst vorteilhaft sind, jedoch die weitere Entwicklung kostspieliger machen können.

### 1.2 Spielidee

Ihr seid Projektleitende in einem Unternehmen. Es ist eure Aufgabe, Softwareprojekte zu planen und durchzuführen. Die Entwickelnden werden das System implementieren und müssen dessen Probleme lösen. Über euch steht das IT-Management und entscheidet über wichtige Angelegenheiten in Sachen Deadlines und Budget. Diese haben viel IT-Hintergrundwissen, aber mit der praxisbezogenen Implementierung haben sie meistens nichts zu tun. Sie wollen in der Lage sein, das



#### Qualität in Software

In der Realität ist den Kunden die Relevanz der Qualität von Software oft nicht bewusst. Trotzdem ist Qualität sehr wichtig für die Weiterentwicklung von Software.






System dauerhaft weiterzuentwickeln und legen daher Wert auf Qualität. Zudem treffen sie Entscheidungen, die sich auf allen Ebenen der Entwicklung auswirken können. Außerdem sind sie in Kontakt mit den Kunden, die möglichst schnell funktionsreiche Software haben wollen.

Das Unternehmen hat gerade zwei neue Kunden an Land gezogen. Die menschlichen und finanziellen Ressourcen sind knapp, weshalb es zurzeit nur Kapazität für ein neues Projekt gibt. Dem Management ist jedoch eine Lösung eingefallen. Ihr als Projektleitende sollt beide Projekte als Prototypen in einer begrenzten Zeit umsetzen. Anschließend wird nur das System tatsächlich umgesetzt, welches einerseits mehr Nutzende generiert hat und andererseits qualitativ hochwertiger ist. Ein gutes Anzeichen für die Qualität liefert das Ausmaß an Technischen Schulden in einem System. Das Ziel ist es also euch gegen das gegnerische Team durchzusetzen und sowohl die Nutzeranzahl als auch die Qualität eures Systems zu steigern.

## 2 Vorbereitung






### 2.1 Spielmaterial

Ihr benötigt folgende Gegenstände, um das Spiel zu spielen:

- 2 Würfel (D6) 
- pro Team eine Spielfigur  
- mind. 30 Spielfiguren einer Farbe zur Verkörperung von Technischen Schulden (TS) 
- mind. 9 Spielfiguren einer anderen Farbe zur Verkörperung des Aufgabenfortschritts 
- Tech-Schulden 2.0 in ausgedruckter Form












Als Spielfiguren eignen sich die klassischen Spielfiguren, Plättchen oder Ähnliches aus einer Spiellesammlung in verschiedenen Farben oder andere kleine identische Gegenstände.

### 2.2 Vor dem ersten Spiel

Zu Beginn müssen alle Spielelemente vorbereitet werden. Dazu müssen alle Karten (  ,  ,  ,  , **Würfelkarten** und **Arbeitsstelle/TS abbauen**) ausgeschnitten werden. Die Seite mit den Übersichten wird einmal in der Mitte durchgeschnitten, sodass jedes Team eine **Übersicht** besitzt. Die **2 Systeme** und die **Nutzeranzeige**  werden nicht ausgeschnitten.

### 2.3 Aufbau

Bereitet die Systeme vor, indem ihr eure Systeme, bestehend aus jeweils zwei Seiten, vor euch zusammenlegt. Platziert neben eurem System die Übersicht zusammen mit den Würfelkarten und der Arbeitsstelle.

In der Mitte des Tisches werden die Karten     verdeckt in jeweils einem Stapel gemischt bereitgelegt. Von diesen werden 3  sowie 1  von den verdeckten Stapeln offen in der Mitte ausgelegt. Neben den Stapeln wird die Nutzeranzeige  platziert. Hierauf werden die Spielfiguren der Teams   auf die 0 gestellt. Zudem werden die Spielfiguren für die TS  und für den Aufgabenfortschritt  in der Mitte bereitgelegt. Das Ganze sollte dann so aussehen wie in *Abbildung 1*.

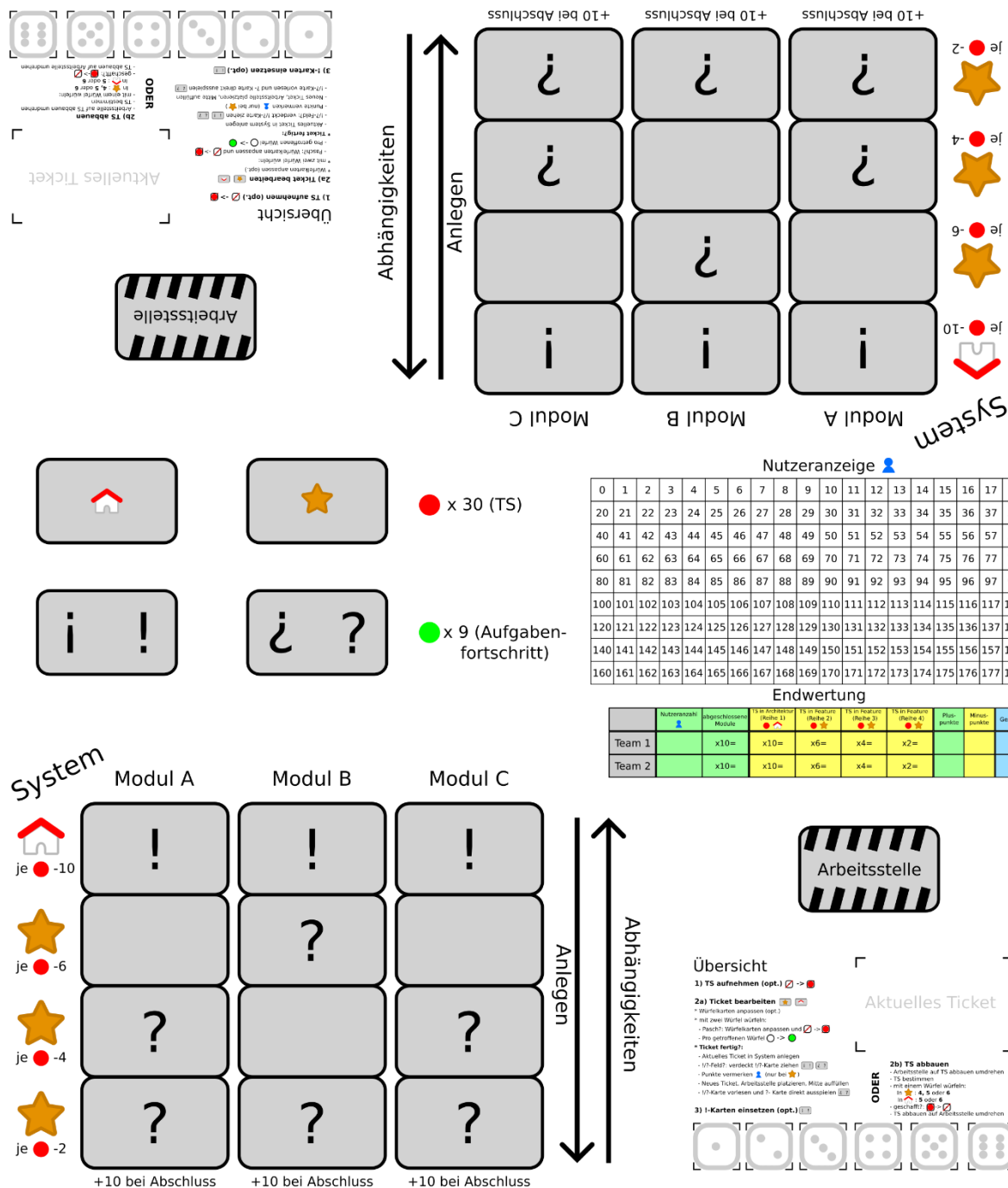




Abbildung 1: Aufbau des Spiels

Vor Beginn der ersten Runde zieht jedes Teams eine  vom verdeckten Stapel und legt es auf den Platz des **aktuellen Tickets** in der **Übersicht**. Platziert anschließend eure **Arbeitsstelle** auf einen beliebigen Platz in der ersten Reihe, dort wo das  abgebildet ist.

### 3 Spielelemente

#### 3.1 Tickets


Tickets sind die einzelnen Komponenten, welche man im Spielverlauf zu einem großen System zusammenbaut. Sie werden nach der Bearbeitung auf einem Feld im System platziert.



Jedes Team arbeitet zu jeder Zeit an genau einem Ticket, welches als **aktuelles Ticket** bezeichnet wird. Es gibt zwei Arten von Tickets.



#### Tickets in der Softwareentwicklung

In der Softwareentwicklung wird oftmals die Ansammlung aller anstehenden Aufgaben in sogenannte Tickets aufgeteilt. Das kann zum Beispiel ein Bugfix, eine neue Funktionalität (Feature), eine neue Organisation der Komponenten (Architektur) oder auch das Abbauen von Technischen Schulden sein. Sie sind also nicht auf die Bereiche Feature und Architektur begrenzt. Die Implementierung eines ganzen Features oder einer Architektur wäre in der Realität zudem in viele kleinere Tickets unterteilt. Tickets entsprechen also eher kleineren Arbeitseinheiten.

• **Architektur**  : Sie verbessern die Infrastruktur und Flexibilität des Systems und ermöglichen so dieses zu vergrößern.

• **Feature**  : Features sind gewünschte Funktionen für das System. Diese Tickets bringen einem bei abgeschlossener Bearbeitung so viele neue Nutzende, wie die Zahl hinter  anzeigt.

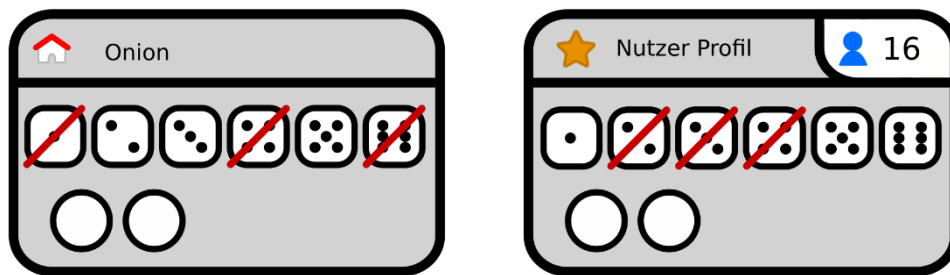








Abbildung 2: Architektur-Ticket und Feature-Ticket






**Aufgaben**  : Aufgaben sind auf den Tickets durch leere kreisförmige Felder gekennzeichnet. Die Anzahl der leeren Felder bestimmt, wie viele Aufgaben bearbeitet werden müssen, um das Ticket zu implementieren. Die Anzahl der Aufgaben kann von 2 bis 5 variieren.

**Freie und gesperrte Würfel**   : Auf den Tickets sind alle Augenzahlen eines Würfels abgebildet. Diese können frei sein  oder gesperrt sein  . Es können 1 bis 5 Augenzahlen auf einem Ticket gesperrt sein.

#### 3.2 System

In eurem System kommen im Laufe des Spiels mehr und mehr Tickets hinzu, wodurch euer System wächst und Nutzende  generiert.

Im Verlauf des Spiels werden Tickets auf verschiedene Felder in das System platziert. Das Symbol auf dem Feld bestimmt, was nach dem Platzieren eines Tickets passiert.

- **leere Felder**  : Hier passiert nichts.
- **Aktionsfelder**  : Es muss eine Aktionskarte  gezogen werden.
- **Ereignisfelder**  : Es muss eine Ereigniskarte  gezogen werden.

### 3.3 Technische Schulden (TS) ●

**TS ●** könnt ihr **bewusst** auf eurem aktuellen Ticket aufnehmen. Das ist vorteilhaft, wann immer ihr möglichst schnell ein Ticket in eurem System anlegen wollt. Ihr könnt TS aber auch **unbewusst** im Laufe des Spiels erhalten. Unabhängig davon, wie TS in das System gelangt sind, bergen diese die Gefahr, dass sie euch im weiteren Spielverlauf hindern können.



#### (Un)bewusste Technische Schulden

Tatsächlich können in der Softwareentwicklung bewusst Technische Schulden aufgenommen werden. Damit werden kurzfristig Abkürzungen für die Entwicklung geschaffen. Das passiert z.B., wenn aus Zeitdruck eine provisorische Lösung implementiert wird. Öfter passiert es, dass Technische Schulden aus diversen Gründen unbewusst in das System gelangen und auch erst später entdeckt werden. Z.B. kann mangelnde fachliche Kompetenz so ein Grund sein. Allgemein besteht bei Technischen Schulden für die zukünftige Entwicklung die potenzielle Gefahr, dass sie zusätzliche Kosten verursachen.

### 3.4 Arbeitsstelle / TS abbauen

Bevor die Bearbeitung eines Tickets begonnen wird, muss entschieden werden, wo das Ticket bei Vollendung an das System angelegt werden soll. An diese Stelle wird die **Arbeitsstelle** gelegt. Die Arbeitsstelle kann auf die Seite **TS abbauen** umgedreht werden, um sich um die Entfernung von Technischen Schulden im System zu kümmern.



Abbildung 3: Arbeitsstelle und TS abbauen

### 3.5 Ereigniskarten und Aktionskarten

Die Karten besitzen alle einen **Namen**, enthalten eine **Beschreibung** und ein fett gedrucktes **Ereignis** bzw. eine **Aktion**.



#### Ursachen und Konsequenzen von Technischen Schulden

Technische Schulden haben so gut wie immer eine Ursache (z.B. Personalmangel oder Zeitdruck). Daraus leiten sich meist auch Konsequenzen ab (z.B. verpasste Deadlines oder Bugs). Ursachen und Konsequenzen können sich miteinander verketten und zu Teufelskreisen entwickeln.

Bestimmte Maßnahmen (z.B. Refactoring) können dabei helfen, Technische Schulden wieder abzubauen oder sie sogar von Anfang an zu verhindern.









- **Ereigniskarten**  : stellen unerwartete Ereignisse während der Entwicklung dar. Sie sind durch einen Buchstaben gekennzeichnet, der die Art eines Ereignisses darstellt: **U** steht für **Ursache** von TS , sie werden zufällig entdeckt und führen direkt zur Aufnahme von TS  im eigenen System. **K** bedeutet **Konsequenz** und kann abhängig von der Anzahl an TS  in eurem System direkte negative Auswirkungen haben. **Je mehr TS  sich also im System befinden, desto höher ist das Risiko für Konsequenzen! Das Ereignis dieser Karte muss direkt durchgeführt werden!**
- **Aktionskarten**  : enthalten Maßnahmen, die für die Entwicklung des Systems hilfreich sein können. Hier darf die **Aktion** in einem vorteilhaften Moment am Ende eines Zugs ausgespielt werden.



Abbildung 4: Ereigniskarte und Aktionskarte

### 3.6 Nutzende





Im Laufe des Spiels könnt ihr **Nutzende**  für euer System bekommen. Diese werden in der Nutzeranzeige  angezeigt.

### 3.7 Übersicht und Würfelkarten



Die neben eurem System liegende **Übersicht** bietet einen Überblick über den Ablauf eines Zugs. Hier sind die einzelnen Schritte der Reihenfolge nach aufgelistet. Zudem werden hier die **Würfelkarten** platziert. Diese bieten einen Überblick, welche Würfel auf dem aktuellen Ticket für die Bearbeitung **frei** und welche **gesperrt** sind. Außerdem ist noch Platz, um das aktuelle Ticket abzulegen.

#### 4 Tickets anlegen und Abhängigkeiten von Tickets

## 4.1 Anlegen von Tickets

Tickets werden nach Abarbeiten aller Aufgaben an das System angelegt. Das System ist dabei in Spalten bzw. Module aufgeteilt und wächst dabei **von oben nach unten**. Dabei dürfen Architektur-Tickets  nur in die erste Reihe des Systems, Feature-Tickets  nur in die darunter liegenden Reihen gelegt werden. Also kommt in jedes Modul zunächst eine Architektur  und anschließend können Features  gelegt werden. Dabei können die Module unabhängig voneinander wachsen.

Wo ein Ticket angelegt werden soll, entscheidet man bereits vor der Bearbeitung eines Tickets, indem die Arbeitsstelle auf das entsprechende Feld im System gelegt wird.

Die *Abbildung 5* zeigt beispielhaft die Anlegemöglichkeiten, also das mögliche Platzieren der Arbeitsstelle, für Tickets in einem fortgeschrittenen System. In diesem Beispiel dürfte eine Architektur  an die blau markierte Stelle und ein Feature  an eine der grün markierten Stellen angelegt werden.

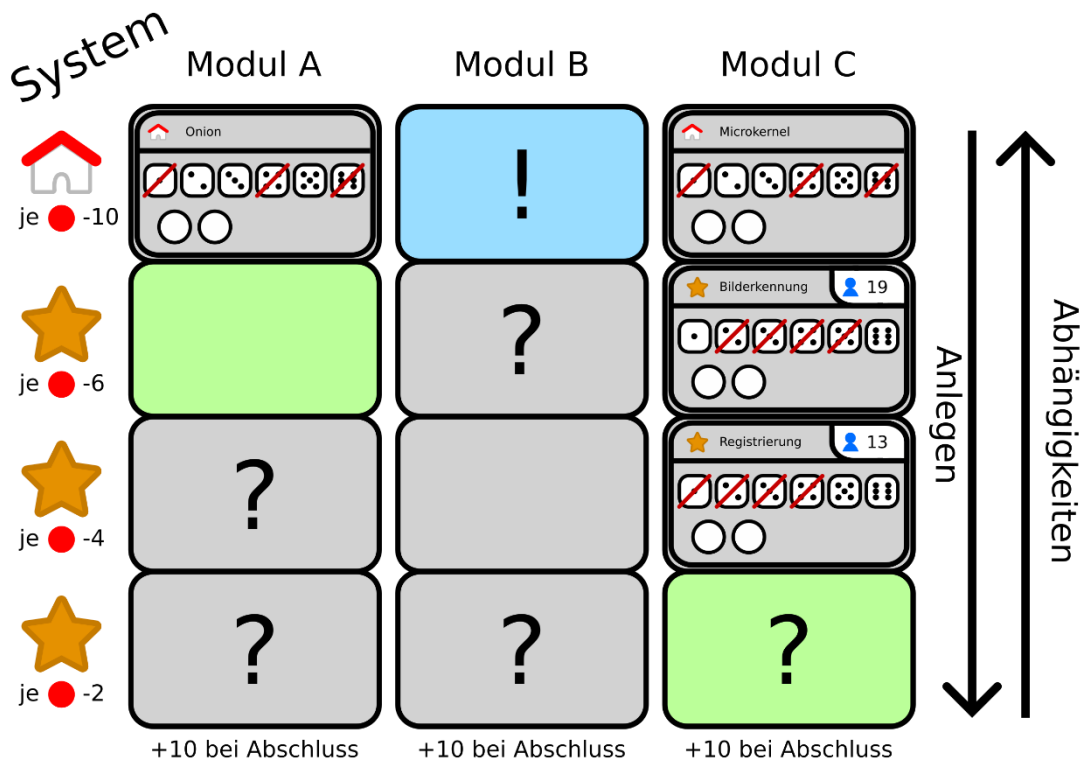




Abbildung 5: Anlegemöglichkeiten für Architektur- und Feature-Tickets



## 4.2 Abhängigkeiten

Ein Ticket ist **immer von allen darüber liegenden Tickets aus demselben Modul abhängig**. Die Abbildungen 6-8 zeigen verschiedene Beispiele für Abhängigkeiten. Dabei steht das Feld mit der Arbeitsstelle und somit das aktuelle Ticket in Abhängigkeit von den orange hinterlegten Tickets.

In *Abbildung 6* soll das aktuelle Ticket in die zweite Reihe von Modul A gelegt werden und ist somit von der Architektur  aus der ersten Reihe von Modul A abhängig.

In *Abbildung 7* soll das aktuelle Ticket in die erste Reihe von Modul B gebaut werden. Dieses Ticket ist nicht abhängig von irgendeinem anderen Ticket, da keine Tickets darüber liegen können. Das gilt für alle Tickets, die in der ersten Reihe gebaut werden, also für alle Architektur-Tickets .

In *Abbildung 8* soll das aktuelle Ticket in die letzte Reihe von Modul C platziert werden. Dieses Ticket ist also von den drei darüberliegenden Tickets abhängig.

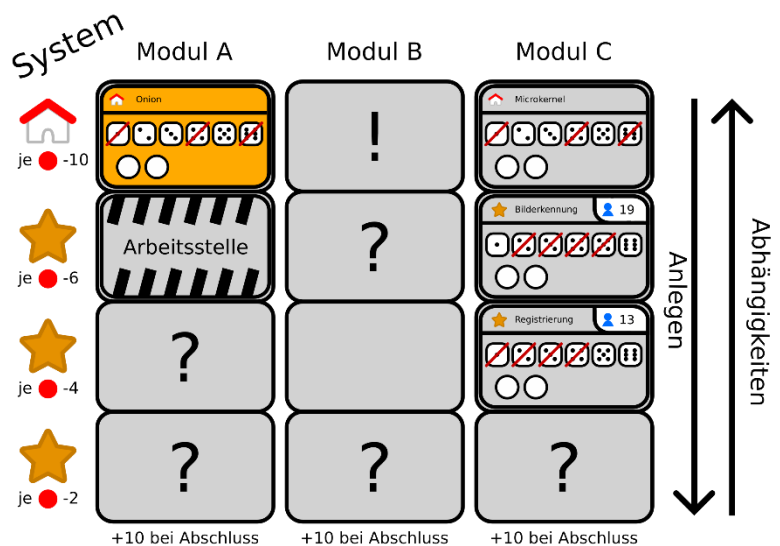


Abbildung 6: Abhängigkeit Beispiel 1

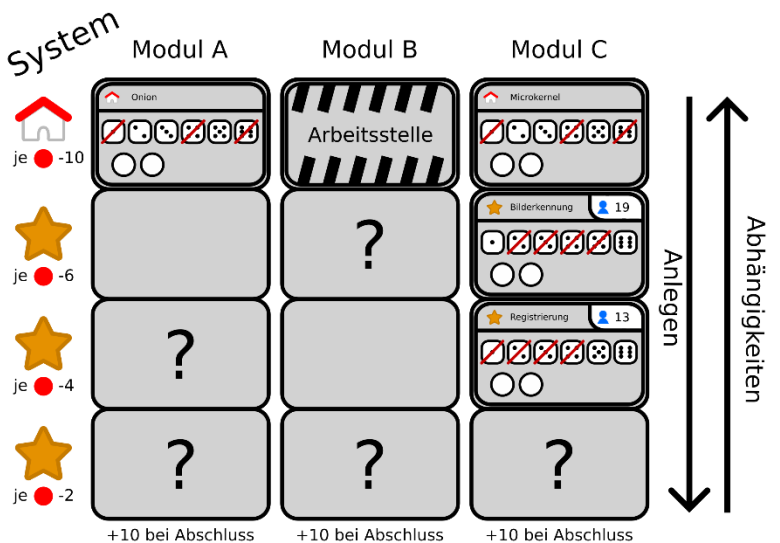


Abbildung 7: Abhängigkeit Beispiel 2

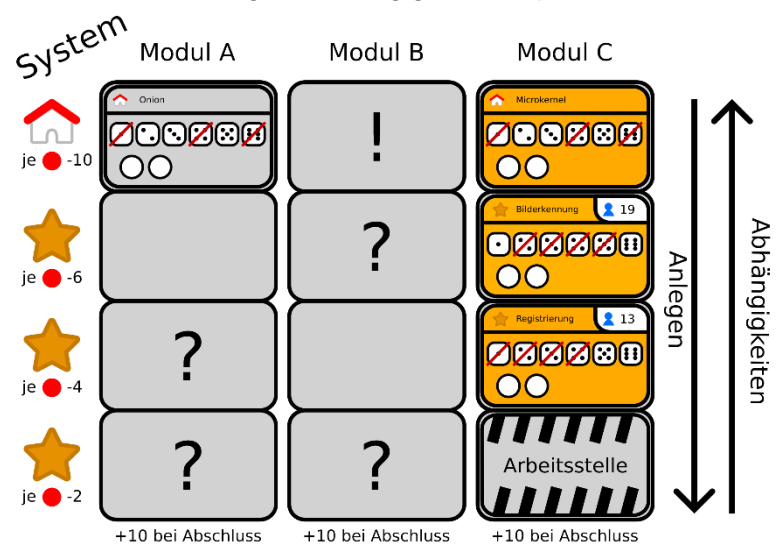





Abbildung 8: Abhängigkeit Beispiel 3



### Abhängigkeiten und Technischen Schulden

Gibt es Technische Schulden in einem System, so können sie potenziell die weitere Entwicklung einschränken. Baut ein Ticket A auf einem mit Technischen Schulden belegtem Ticket B auf, so kann es passieren, dass die Entwicklung für das Ticket A eingeschränkt ist oder sogar unmöglich ist. Dieses Problem kann sich auf weitere abhängige Tickets auswirken und weitere Technische Schulden verursachen. Dabei beschränken sich Abhängigkeiten in der Realität nicht zwingend auf Tickets, sondern können auch beispielsweise ganze Module betreffen.



Ist das aktuelle Ticket von einem anderen Ticket abhängig, so sind die Augenzahlen, auf denen das andere Ticket eine TS  hat, für die Bearbeitung des aktuellen Tickets gesperrt. Diese Sperrung kann **nicht** durch die (un-)freiwillige Aufnahme einer TS  auf der entsprechenden Augenzahl auf dem aktuellen Ticket umgangen werden. Die gesperrten Augenzahlen ohne TS  von bereits verbauten Tickets haben keinen Einfluss auf das aktuelle Ticket.

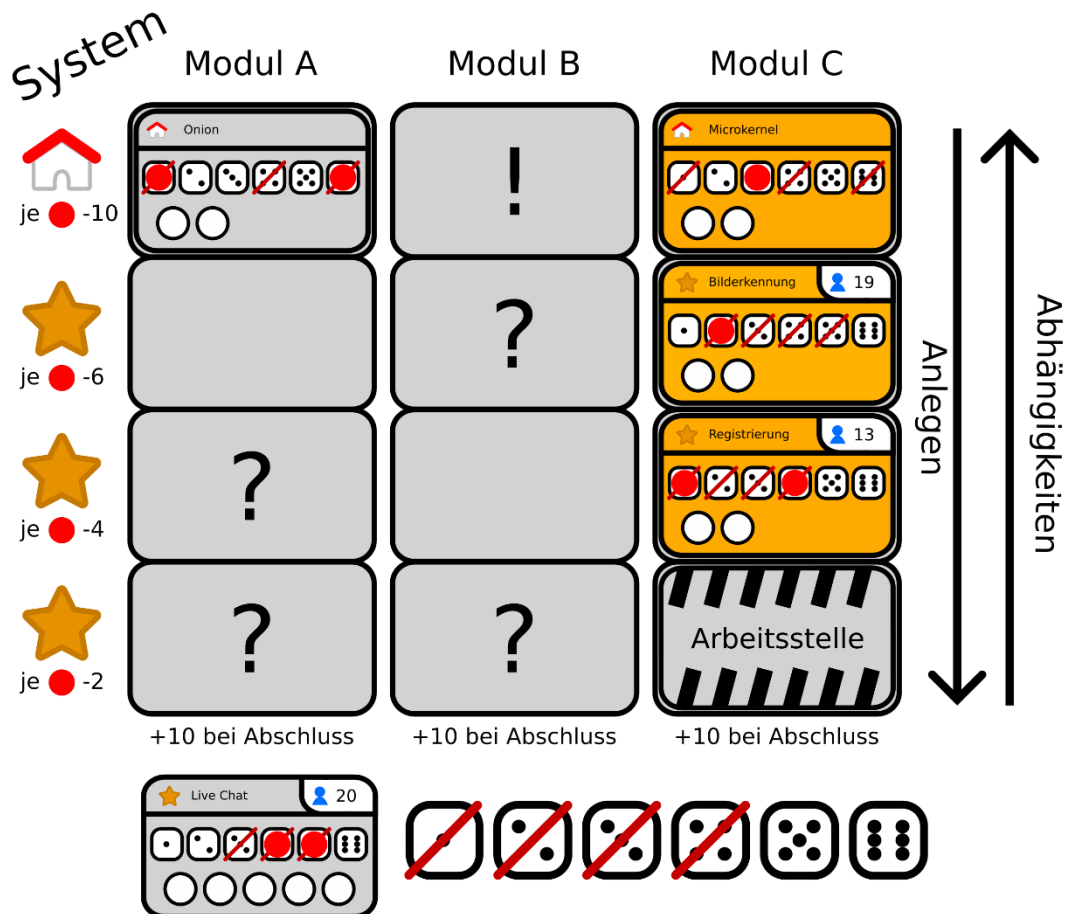







Abbildung 9: Abhängigkeit mit TS

Die *Abbildung 9* zeigt ein Beispiel für Abhängigkeiten mit TS  im System und die sich daraus ergebenden freien  und gesperrten Würfel  für die Bearbeitung der Aufgaben. Hier soll das aktuelle Ticket **Live Chat** auf das Feld der Arbeitsstelle in das System implementiert werden. Das Ticket ist von den darüber liegenden Features  **Bilderkennung** und **Registrierung** sowie der Architektur  **Microkernel** abhängig. Alle anderen Tickets haben auf das aktuelle Ticket keinen Einfluss.

Die **gesperrten** und **freien** Würfel ergeben sich wie folgt:

- 1 ist wegen der TS auf dem Ticket Registrierung gesperrt.
- 2 ist wegen der TS auf dem Ticket Bilderkennung gesperrt.
- 3 ist sowohl wegen der TS auf dem Ticket Microkernel als auch wegen der gesperrten Augenzahl auf dem aktuellen Ticket gesperrt.
- 4 ist trotz der durch einen Pasch unbewussten aufgenommenen TS auf dem aktuellen Ticket wegen der TS auf dem Ticket Registrierung gesperrt.
- 5 ist wegen der bewusst aufgenommenen TS auf dem aktuellen Ticket frei.
- 6 ist wegen des freien Würfels auf dem aktuellen Ticket frei.

## 5 Spielverlauf

Um das Softwaresystem zu bauen, werden mehrere Runden gespielt. In einer Runde macht jedes Team einen Spielzug. Ein Zug ist dabei in drei Schritte unterteilt.

### 1) TS aufnehmen (optional) ●

- Zunächst könnt ihr absichtlich beliebig viele TS ● auf eurem aktuellen Ticket auf **gesperrte Würfel** ☒ aufnehmen. Das passiert, indem ihr eine TS ● auf dem gewünschten Würfel eines Tickets platziert ☒. Anschließend könnt ihr diesen als

**freien Würfel** ☐ zur Bearbeitung von Aufgaben nutzen, solange dieser nicht durch eine TS ● eines **abhängigen Tickets** gesperrt ist. Damit geht das Implementieren des aktuellen Tickets deutlich schneller. Doch aufgepasst, später können **andere Tickets, die abhängig von diesem Ticket sind, die mit TS belegten Würfel ☒ nicht mehr zum Bearbeiten des aktuellen Tickets nutzen!**



#### Aufnehmen von Technischen Schulden

Die bewusste Aufnahme von Technischen Schulden führt dazu, dass Tickets schneller bearbeitet werden können. Für zukünftige Tickets, die auf diesem Ticket aufbauen, bedeutet es allerdings, dass sie langsamer bearbeitet werden können.

### 2) Ticket bearbeiten ODER TS abbauen

- In diesem Schritt müsst ihr euch entscheiden, ob ihr entweder euer aktuelles Ticket bearbeitet oder ob ihr TS in eurem System abbauen wollt.

**2a) Ticket bearbeiten:** Um euer aktuelles Ticket zu bearbeiten, sind folgende Schritte notwendig:

#### – Würfelkarten aktualisieren:

- Zunächst ist es hilfreich, die eigenen Würfelkarten zu aktualisieren. Ihr schaut also, welche Würfel für das aktuelle Ticket frei ☐ und welche gesperrt ☒ sind. Das ist allerdings **nicht immer nur auf dem aktuellen Ticket selbst ersichtlich, da die Bearbeitung eines Tickets ja auch von anderen Tickets abhängig ist!** Die Würfelkarten könnt ihr auch jederzeit anpassen, das kann helfen, wenn ihr euch eine Strategie überlegen wollt. Erfahrene Spieler können diesen Schritt auch überspringen.

#### – Würfeln:


















- Nun würfelt ihr mit **zwei** Würfeln. Solltet ihr dabei einen **Pasch** gewürfelt haben, bekommt ihr automatisch eine TS ● auf die Zahl mit der doppelt gewürfelten Augenzahl, solange dort noch keine liegt. Dabei ist es egal, ob der Würfel auf dem aktuellen Ticket gesperrt ☒ oder frei ☐ ist. Wenn diese Augenzahl durch keine TS ● eines **abhängigen Tickets** gesperrt ist, kann der gewürfelte Pasch direkt zur Bearbeitung des aktuellen Tickets genutzt werden.
- Anschließend wird geschaut, wie viele freie Würfel ☐ beim Würfeln getroffen wurden. Platziert pro getroffenen freien Würfel ☐ eine Spielfigur ● auf ein leeres Aufgabenfeld. Wenn die Aufgabenfelder bereits voll sind, verfällt der überschüssige Würfel.





#### Unvermeidbare Technischen Schulden

Die Aufnahme von Technischen Schulden kann man nicht komplett vermeiden, weil sie auch unbewusst aufgenommen werden können. Dafür führt dies aber, wenn auch unbeabsichtigt, zu Abkürzungen in der Entwicklung.

### – Ticket fertig?:




- Sind alle Aufgaben abgearbeitet, wird das aktuelle Ticket in das System integriert. Entfernt die Spielfiguren  für die Bearbeitung der Aufgaben. **Die TS  bleiben auf dem Ticket!** Tauscht anschließend die Arbeitsstelle mit dem aktuellen Ticket. Falls das Ticket auf einem **Ereignisfeld**  oder **Aktionsfeld**  gelandet ist, nehmt euch eine entsprechende **Ereigniskarte**  bzw. **Aktionskarte**  und legt sie verdeckt bereit, aber **schaut sie euch noch nicht an!**
- Falls ihr ein Feature  implementiert habt, vermerkt die generierte Nutzeranzahl  in der Nutzeranzeige .
- Entscheidet euch danach für ein neues aktuelles Ticket aus der Mitte (Feature  oder Architektur ) und legt die Arbeitsstelle an eine gültige Position in eurem System. Achtet darauf, dass in der Mitte des Tisches immer 3 Features  und 1 Architektur  zur Auswahl bereitliegen.
- Falls ihr eine  oder  gezogen habt, dürft ihr sie nun anschauen. Lest die Karte laut vor. Die **Ereigniskarte**  müsst ihr **direkt ausführen**. Die **Aktionskarte**  darf in einem **vorteilhaften Moment** am Ende eines Zuges ausgespielt werden.


**2b) TS abbauen:** In diesem Schritt könnt ihr die in eurem System angesammelten TS  abbauen. Zunächst dreht ihr eure Arbeitsstelle auf die Seite **TS abbauen** um. Dann entscheidet ihr euch für eine bestimmte TS , die abgebaut werden soll. Anschließend wird mit **einem** Würfel gewürfelt.





#### Abbauen von Technischen Schulden (in Architektur)

Hat man sich erst mal für das Abbauen von Technischen Schulden entschieden, so kann diese Aufgabe sehr im Umfang variieren. Je mehr Tickets von einer Technischen Schuld abhängig sind, desto aufwändiger ist es sie wieder zu entfernen. Architekturschulden sind dabei in der Regel am schwierigsten zu entfernen, da Architektur meistens sehr zentral in Systemen ist und vieles darauf aufbaut. Gerade diese Art von Technischen Schulden sollten gut unter Kontrolle gebracht werden.

- Um eine beliebige TS  in einem **Feature**  abzubauen, müsst ihr eine **4, 5** oder **6** würfeln.
- Um eine beliebige TS  in einer **Architektur**  abzubauen, müsst ihr eine **5** oder **6** würfeln.


Habt ihr die passende Zahl gewürfelt, dann nehmt die entsprechende TS  von eurem Ticket. Ist euch das nicht gelungen, könnt ihr im nächsten Spielzug wieder entscheiden, ob ihr Schritt 2a) oder 2b) ausführt. Dreht die Karte TS abbauen wieder auf die Seite der **Arbeitsstelle**.

### 3) Aktionskarten einsetzen (optional)

- In diesem Schritt könnt ihr beliebig viele bereits gezogene Aktionskarten  ausspielen. Das bedeutet, ihr führt die fett gedruckte Aktion aus. Anschließend wird die Karte beiseitegelegt. Dies ist nur am Ende eures Spielzugs möglich. Wenn ihr keine Aktionskarten  besitzt, wird dieser Schritt übersprungen.

In Kurzfassung findet ihr die einzelnen Schritte eines Spielzugs auch auf der **Übersicht**.

## 6 Spielende

Das Spiel endet entweder so bald in einem System alle 12 Plätze für Tickets belegt wurden oder nach einer vorher festgelegten Zeit. In jedem Fall wird die Runde noch zu Ende gespielt. Anschließend werden zu den bereits gesammelten Nutzenden  für jedes System folgende Punkte verrechnet:





### Technische Schulden in Architektur




Technische Schulden sind besonders dann kritisch, wenn vieles im System auf ihnen aufbaut, z. B. bei der Architektur. Aber auch Features können beispielsweise sehr zentral im System sein, wodurch Technische Schulden hier große Auswirkungen haben können. Lediglich Schulden, die sich ganz außen im System befinden und somit keine Abhängigkeiten verursachen, sind potenziell weniger gefährlich.

- für jedes abgeschlossene **Modul +10 Punkte**

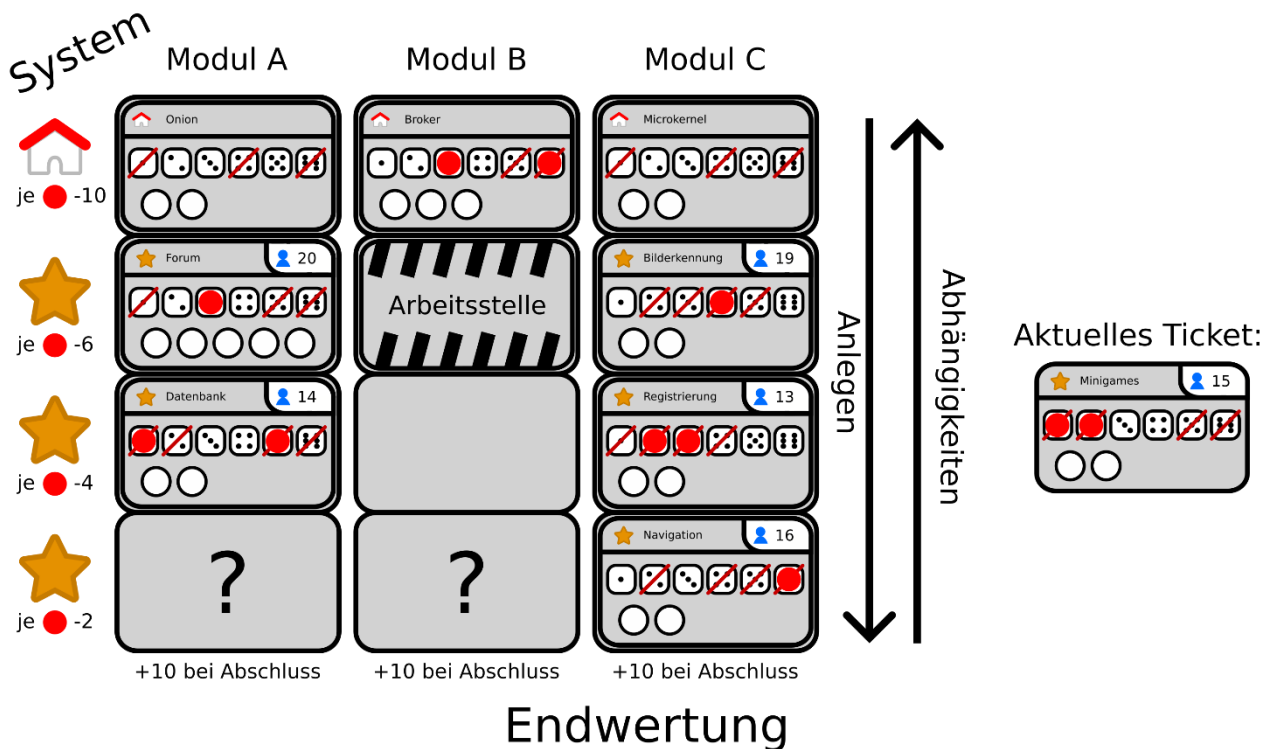
- für jede TS  auf **Architektur**  **-10 Punkte**

- für jede TS  auf **Feature**   
in Reihe 2 / 3 / 4 je **-6 / -4 / -2 Punkte**

Sowohl die TS  als auch die Nutzeranzahl  **vom aktuellen Ticket gehen nicht mit in die Endwertung ein!**

Tragt die einzelnen Posten am besten in der Tabelle für die **Endwertung** ein, die sich unter der **Nutzeranzeige**  befindet. Die gesammelte Nutzeranzahl  könnt ihr direkt aus der Nutzeranzeige  entnehmen. Die *Abbildung 10* zeigt ein Beispiel zur Berechnung der Endwertung.

Das Team mit den **meisten Punkten** hat den besseren Prototypen gebaut und gewinnt das Spiel.



	Nutzeranzahl 	abgeschlossene Module	TS in Architektur (Reihe 1) 	TS in Feature (Reihe 2) 	TS in Feature (Reihe 3) 	TS in Feature (Reihe 4) 	Plus- punkte	Minus- punkte	Gesamt
Team 1	82	1x10=10	2x10=20	2x6=12	4x4=16	1x2=2	92	50	42

Abbildung 10: Beispiel für die Endrechnung

### Beispielrechnung:

In diesem Beispiel hat das Team 82 Nutzende generiert. Zudem hat es das Modul C abgeschlossen, was 10 Punkten entspricht. Aus diesen beiden Zahlen ergeben sich die **92 Pluspunkte** des Teams.

Nun werden anhand der Anzahl von TS Minuspunkte berechnet. Das Team hat insgesamt 2 TS auf Architektur-Tickets . Diese werden mit 10 multipliziert, woraus sich dann 20 Minuspunkte für die Architektur ergeben.

Die TS in Feature-Tickets werden danach unterteilt, in welcher Reihe das jeweilige Ticket mit einer TS liegt. In der zweiten Reihe bekommt jede TS je 6 Minuspunkte. In diesem Beispiel sind das je eine TS auf den Tickets Forum in Modul A und Bilderkennung in Modul C, also 12 Minuspunkte. Die Features Datenbank in Modul A und Registrierung in Modul C liegen in der dritten Reihe, wo jede TS 4 Minuspunkten entspricht. Hier liegen 4 TS und somit ergeben sich 16 Minuspunkte. Schließlich gibt es noch 2 Minuspunkte pro TS in der vierten Reihe. Hier liegt nur das Feature Navigation in Modul C, welches eine TS hat und somit 2 Minuspunkte generiert.

Somit kommt man auf **50 Minuspunkte**. Für das Endergebnis werden die Minuspunkte von den Pluspunkten abgezogen. Insgesamt hat das Team in diesem Beispiel also **42 Punkte** erreicht.