

Scaled Agile: Toolgestützte Echtzeitplausibilisierung des PI-Planning

Ulrich Vogl¹, Markus Siegle²

Abstract: Agiles Projektvorgehen mit seinen fortschrittlichen Methoden hat sich mittlerweile weitgehend durchgesetzt. So schlank einzelne agile Teams agieren, so komplex wird das Zusammenspiel mehrerer oder gar vieler solcher Teams - insbesondere im Rahmen der auch hier erforderlichen Planung. Aktuelle skalierende Ansätze wie SAFe (Scaled Agile Framework) bieten dafür zwar grundsätzliche Verfahren, jedoch zeigt die Praxis, dass man auf der Meta-Ebene mit zunehmender Anzahl der beteiligten Teams meist zu optimistisch unterwegs ist: Spätestens nach der Auflösung Team-übergreifender Abhängigkeiten ist beim PI-Planning die Komplexität in der Regel so hoch, dass eine realistische Gesamteinschätzung der Leistbarkeit über alle Teams und das gesamte Program Increment (PI) nicht mehr möglich ist.

Das von uns entwickelte Tool ist in der Lage, für beliebige Story-on-Node-Netzwerke des agilen PI-Plannings Wahrscheinlichkeitsverteilungen abzuleiten und daraus Gesamtwerte zu berechnen, um schließlich aussagekräftige Metriken für Leistbarkeit und Qualität abzuleiten. Im Gegensatz zu bekannten Ansätzen basiert dies nicht auf Simulation, sondern auf der exakten numerischen bzw. approximierten Model Checking-Berechnung der resultierenden Wahrscheinlichkeitsverteilungen, womit typische PI-Planungen sehr effizient durchgerechnet und bewertet werden können.

Keywords: SAFe, PI-Planning, Toolgestützte Echtzeit-Plausibilisierung

1 Einleitung

Agiles Projektvorgehen ist heute nicht mehr wegzudenken und hat - dank skalierender Ansätze - auch bei großen IT-Projekten mittlerweile flächendeckend Einzug gehalten. Doch während das methodische Vorgehen innerhalb eines einzelnen Scrum-Teams bekannter Maßen auf einen „Bierdeckel“ passt, wird es bei der Vernetzung der Aktivitäten mehrerer Teams schnell komplex. Aktuelle skalierende Ansätze wie SAFe (*Scaled Agile Framework*) bieten zwar auch hierfür Lösungsansätze, jedoch werden Planungsqualität und -zeitreue auf höherer Planungsebene (*Agile Release Train*, ART und *Program Increment*, PI) mit zunehmender Anzahl der Teams schwierig handhabbar. Es ist wohl der Einfachheit dieser Ansätze geschuldet, dass die Ziele im realen Projektleben meist zu optimistisch geplant sind und die Ergebnisse der System Demos dann regelmäßig signifikant hinter den gesetzten Definitions-of-Done (DoD) zurück bleiben: speziell bei harten Abhängigkeiten, z.B. bei fix gebuchten Ressourcen oder bei anstehenden Rollout-Terminen mit vielen Beteiligten, oftmals hoch problematisch.

¹ Universität der Bundeswehr München, Institut für Technische Informatik,
Werner-Heisenberg Weg 39, D-85577 Neubiberg, ulrich.vogl@unibw.de

² Universität der Bundeswehr München, Institut für Technische Informatik,
Werner-Heisenberg Weg 39, D-85577 Neubiberg, markus.siegle@unibw.de

Grundsätzlich basiert die PI-Planung bei SAFe auf jeweils teaminternen Feinplanungen der (User-)Storys für alle Sprints eines PI und unter Maßgabe der Team-verfügbaren Ressourcen ("Story Point Poker"). Darüber hinaus werden in einem zweiten Schritt Abhängigkeiten zwischen den Storys verschiedener Teams identifiziert und schließlich aufgelöst. Wie erwähnt, legt die sich daraus ergebende hohe Komplexität für eine belastbare übergreifende Plausibilisierung des PI-Planning nun eine Tool-Unterstützung nahe.

Das hier vorgestellte Tool kann grundsätzlich für beliebige in Projekten typischer Weise auftretende Activity-On-Node-Graphen verwendet werden und dort auf Basis probabilistischer Einzel-Laufzeitverteilungen Gesamt-Laufzeitverteilungen berechnen. So auch für den hier beschriebenen Usecase eines teamübergreifenden Story-On-Node-Graphen im PI-Zyklus. Aus dem Ergebnis des *Story Point Poker* und der verfügbaren Ressourcenausstattung des Teams bestimmen wir zuerst je Einzelstory eine Wahrscheinlichkeitsverteilung der Laufzeit. Zusammen mit den Präcedenzen der Storys wird daraus eine exakte bzw. approximierte Gesamtverteilung berechnet und über Verteilungsfunktion und Quantile entsprechende Metriken für die Einschätzung des gesamten PI abgeleitet.

Während andere Arbeiten vergleichbare Probleme oftmals simulativ behandeln, ist unser Tool deutlich effizienter unterwegs, denn die Zielverteilungen werden nun analytisch bestimmt: entweder numerisch oder (approximiert) per Model Checking. Erschwingliche Hochleistungsprozessoren (z.B. AMD RYZEN-Reihe) ermöglichen solch aufwändige Berechnungen heute bereits mit handelsüblichen PCs - unterhalb 1000€ Anschaffungskosten. Konkret rechnen wir serienparallele Konstellationen stets durch numerische Faltung bzw. Maximumsbildung [1]. Die Einzelverteilungen komplexer Konstrukte hingegen werden zuerst über Summen zweier Erlang-Verteilungen approximiert („Fitting“) und die resultierende Gesamtverteilung schließlich über probabilistisches Model Checking [2] (mit Hilfe des Tools PRISM [9]) ermittelt. Als Metriken haben wir exemplarisch die *Wahrscheinlichkeit einer zeit- und DoD-treuen Implementierung des PI* sowie den *Über-/Unterplanungsgrad der PI bei einer 50% bzw. 90%-Konfidenz* implementiert.

Der Rest des Papers ist wie folgt aufgebaut: Kapitel 2 führt kurz in wichtige SAFe-Begrifflichkeiten ein, während Kapitel 3 konkret die von uns entwickelten Lösungsstrategien vorstellt. In Kapitel 4 werden diese schließlich anhand eines Fallbeispiels praktisch erläutert. Kapitel 5 rundet mit Schlussfolgerungen und Ausblicken ab.

2 PI-Planning im SAFe

Skalierende agile Ansätze nutzen für mehrere Teams bzw. mehrere Sprints klammernde Strukturen und Methoden. Im Falle von SAFe sind das der ART bzw. das PI [10, 8]: Ein PI bildet die zeitliche Klammer über mehrere Sprints und stellt somit den Planungshorizont dar. Es umfasst zeitlich typischer Weise 8-12 Wochen plus die Planung des nächsten PI. Diese findet stets am PI-Ende, in Form eines Meetings aller PI-beteiligten Teams (nämlich des ART) statt, wobei die Sprints des nachfolgenden PI - feingranular auf Ebene einzelner Storys geplant werden. Dies geschieht wie folgt:

2.1 Storypoint-Poker

Im ersten Schritt priorisiert jedes Team sein Backlog und legt fest, welche Storys davon in welcher Abfolge im nächsten PI angegangen werden sollen (Storyliste). Um eine Größenordnung der inhaltlichen und zeitlichen Komplexität zu erhalten, werden pro Story die *Storypoints* (SP) in Form adaptierter Fibonacci-Zahlen (1, 2, 3, 5, 8, 13, 20, 40, 100) ermittelt. Dies erfolgt methodisch im sog. *Storypoint-Poker*.

2.2 Ressourcen-Verfügbarkeit

Um eine Größenordnung für die Fibonaccizahlen des Storypoint-Poker zu erhalten, normiert SAFe das Leistungsspensum einer Vollzeitkraft in zwei Arbeits-Wochen (10 Personentage, PT) auf 8 SP, was einem Faktor **1,25 PT/SP** entspricht. Nach diesem Verfahren werden nun für jeden Sprint des PI die aus den Ressourcen abgeleiteten verfügbaren SP ermittelt. Schließlich wird die zuvor ermittelte Storyliste pro Sprint gegen die verfügbaren *Storypoints* gelegt und damit determiniert, welche Story in welchem Sprint mit wieviel Ressourcenbeteiligung (Mitarbeiterkapazität, MAK) abgearbeitet wird.

2.3 Wahrscheinlichkeitsverteilung für die Story-Laufzeit

Die zuvor aufgeführte SAFe-Methodik bildet eine geeignete Basis, pro geschätzter Story anhand der Storypoints und der beteiligten Ressourcen eine Wahrscheinlichkeitsverteilung zu finden, die die zeitliche Ausdehnung der Abarbeitung widerspiegelt. Dabei soll einfließen, dass mit steigender Ressourcenbeteiligung zwar in erster Ordnung die zeitliche Ausdehnung der Story proportional abnimmt, aber in zweiter Ordnung wegen des erhöhten Abstimmbedarfs (**10%** des eigentlichen Aufwands) wieder zunimmt. Außerdem sinkt im gleichen Zuge das Risiko einer schwer handhabbaren bzw. unerwarteten Komplexität, welches sich in der Standardabweichung widerspiegelt.

Mit den beiden Konstanten ergibt sich für den Erwartungswert der Story-Laufzeit:

$$\mu_{Story} = SP \cdot 1,25 \cdot \frac{1 + 10\% \cdot (MAK - 1)}{MAK}$$

Die Standardabweichung wählen wir anteilig vom Erwartungswert wie folgt: Pauschal 25% davon als sachthemen-kausales Risiko sowie 50% dividiert durch die Anzahl der beteiligten Ressourcen (AnzRes) als Kompetenz-Risiko.

$$\sigma_{Story} = \mu_{Story} \cdot (0,25 + 0,5/AnzRes)$$

Als Wahrscheinlichkeitsverteilung haben probabilistische Projektplanungsmethoden wie PERT [4] in den vergangenen Jahrzehnten oftmals die *verallgemeinerte β -Verteilung* herangezogen, da sich hiermit insbesondere die für Projektlaufzeiten in der Praxis typischen linksschiefen Verteilungen gut abbilden lassen. Goldratt [6] führt hierfür drei Prinzipien an: *Student-Syndrom*, *Parkinson's Law* sowie *Murphy's Law*. Lässt man diese außen vor, wäre die *Normalverteilung* die wohl naheliegende, so dass wir diese (der Einfachheit halber) in unserem Beispiel auch verwenden wollen.

2.4 PI-Präzedenzgraph

Aus den gegebenen Präzedenzen der Storys, die sich in Form von einfachen Vorgänger-Nachfolger-Beziehungen beschreiben lassen, und den soeben ermittelten Story-individuellen Wahrscheinlichkeitsverteilungen kann nun ein Präzedenzgraph erstellt werden, an dem zusätzlich jeder Knoten die Parameter seiner Wahrscheinlichkeitsverteilung darstellt. Damit sind alle Voraussetzungen für die Benutzung unseres Tools gegeben.

3 Lösungsstrategien im Tool

3.1 Theoretischer Hintergrund und Implementierungsansatz

Unser Tool bestimmt die Wahrscheinlichkeitsverteilung der Gesamtlaufzeit von Präzedenzgraphen bis zu einer gewissen Größe/Komplexität auf effiziente Weise. Die Strategie besteht darin, mit numerischen Dichtefunktionen mit anpassbarer Schrittweite (aus der sich die Anzahl der zu verarbeitenden Stützstellen ergibt) zu arbeiten. Der Ausgangsgraph wird iterativ auf schließlich einen einzigen Knoten reduziert, wobei drei Möglichkeiten der Reduktion zum Einsatz kommen:

- 1) Zwei seriell verknüpfte Knoten des Graphen werden zu einem einzigen verschmolzen, dessen Dichte sich durch numerische Faltung der beiden Operanden-Dichten ergibt. Dabei sind zwei Knoten A und B seriell verknüpft, wenn A der einzige Vorgänger von B und B der einzige Nachfolger von A ist [7].
- 2) Eine Menge parallel verknüpfter Knoten wird auf einen einzigen Knoten reduziert, dessen Dichte sich durch numerische Maximum-Berechnung der beteiligten Knoten-Verteilungen ergibt. Dabei gelten Knoten als parallel verknüpft, wenn sie alle genau dieselben Vorgänger und genau dieselben Nachfolger haben [7].
- 3) Falls an einem Punkt der Reduktion weder 1) noch 2) anwendbar sind, findet unser Tool mit einem speziell dafür entwickelten Algorithmus sog. minimale komplexe Cluster, die jeweils durch einen einzigen Ersatzknoten zu ersetzen sind. Ein komplexes Cluster ist ein Teilgraph mit einem einzigen Einstiegs- und einem einzigen Endpunkt (im Sinne eines Synchronisationspunktes). Um ein komplexes Cluster quantitativ zu analysieren, wird es in ein PRISM-Modell transformiert [9]. Der Model Checker PRISM liefert dann die Werte der Verteilungsfunktion der Bearbeitungszeit des Clusters, die anschließend als numerische Verteilung des Ersatzknotens in die weitere Analyse einfließen. Gegenüber der seriellen bzw. parallelen Reduktion ist die PRISM-basierte Analyse komplexer Cluster wesentlich zeitaufwändiger, weshalb dieser Schritt so selten wie möglich und immer für möglichst kleine Cluster ausgeführt wird.

Details zur numerischen Analyse stochastischer Graphmodelle sind in unserer früheren Arbeit [11] ausgeführt. Bei den numerischen Berechnungsverfahren ermöglicht die Wahl der Stützpunktzahl einen individuell anpassbaren Tradeoff zwischen Genauigkeit und Rechenaufwand.

Der Model Checker PRISM arbeitet intern nur mit Exponentialverteilungen, d.h. allgemeine Verteilungen müssen durch Phasenverteilungen, bestehend aus exponentiellen Anteilen (sog. Phasen), approximiert werden, damit PRISM damit umgehen kann. Wir haben uns entschieden, für das Fitting einer gegebenen Verteilung (mit Variationskoeffizient kleiner als 1) eine Sequenz zweier unabhängiger Erlang-Verteilungen zu benutzen, wobei dieser Ansatz durch den Austausch des Fitting-Moduls unseres Tools leicht verallgemeinert werden könnte. Eine Erlang-Verteilung ist die Summe von k unabhängig identisch verteilten Exponentialverteilungen (jede davon mit Parameter λ), d.h. für das Fitting mit einer Sequenz zweier Erlang-Verteilungen gilt es, vier Parameter $(k_1, \lambda_1, k_2, \lambda_2)$ zu bestimmen. Für eine vorgegebene Wahrscheinlichkeitsverteilung bestimmt unser Tool diese vier Parameter derart, dass Erwartungswert und Varianz exakt getroffen werden und die Abweichung des dritten Moments minimiert wird.

Schließlich noch ein Wort zur Frage, wie das Tool mit der Beschränkung von (Personal-) Ressourcen umgehen kann: Es ist nämlich auch in der Lage, den Ressourcen-Bedarf von gleichzeitig aktiven Knoten zu analysieren und bei jeder Überschreitung der verfügbaren Ressourcen zusätzliche Kanten (also Abhängigkeiten) in den Präzedenzgraphen einzufügen, d.h. bestimmte Aufgaben werden serialisiert. Dabei folgen wir der Strategie von Bell & Han [3]. Allerdings musste dafür deren Algorithmus von einem deterministischen in unser stochastisches Rahmenwerk übertragen werden.

3.2 Ziel-Metriken

Unser Tool ist durch die im vorigen Abschnitt beschriebenen Berechnungen in der Lage, die Wahrscheinlichkeit einer zeit- und DoD-treuen Implementierung des PI zu bestimmen. Aus der Gesamt-Verteilungsfunktion lässt sich unmittelbar die Wahrscheinlichkeit bestimmen, dass das PI in der vorgegebenen Zeit abgeschlossen werden kann.

Darüber hinaus ergeben sich aus den Quantilen der Gesamt-Verteilungsfunktion der Über- bzw. Unterplanungsgrad eines PI. Bei vorgegebener Zeit-Zielmarke t_1 und berechnetem p -Quantil t_p (zur Konfidenz p , also z.B. für $p = 90\%$) ergibt sich der Über- bzw. Unterplanungsgrad aus dem Quotienten t_p/t_1 . Bei einer guten Planung sollte dieser Quotient in der Nähe von 1 liegen, Werte wesentlich größer als 1 zeigen eine Überplanung an, d.h. die Zielmarke ist mit den eingeplanten Ressourcen nicht erreichbar.

4 Fallbeispiel

Nach der theoretischen Abhandlung wollen wir den Nutzen des vorgestellten Tools nun exemplarisch anhand eines fiktiven, aber realitätsnahen agilen Großprojektes darstellen: *Bei einem Finanzdienstleister soll die IT-Anwendungslandschaft für die Unternehmenssteuerung erneuert und in diesem Zuge eine Standardsoftware eingeführt werden. Das Projektvorgehen ist agil unter Nutzung des SAFe-Frameworks.*

4.1 PI-Struktur

Zeitlich soll sich ein PI über 14 Wochen erstrecken, in denen 3 Sprints a 4 Wochen plus eine Replanning-/Kickoff-Phase (2 Wochen) Platz finden. Letztere umfasst die ART-weite Planung und den Start des nächst nachfolgenden PIs. Grundsätzlich wird mit 20 Arbeitstagen pro Sprint (bzw. 10 für die Replanning-/Kickoff-Phase) gerechnet, so dass sich eine Gesamtlänge von 70 Arbeitstagen für das PI ergibt.

4.2 Agile Projektstruktur: ART mit 3 agilen Teams

Grundsätzlich ist unser Projektbeispiel stark vereinfacht, bestehend aus einem ART mit nur drei untergeordneten Scrum-Teams. Jedes Team soll dabei über zwei Vollzeitressourcen (A&B, C&D bzw. E&F) verfügen. Der ART hat zusätzlich für Planungs- und Synchronisationsaufgaben die übergeordnete Vollzeitressource „Product Manager“ (G).

Projektorganigramm (vereinfacht), bestehend aus den agilen Teams:

- 100 (Finanzprodukte Nebenbuch, inkl. Risikoversorge)
- 200 (Hauptbuch und Konzernkonsolidierung)
- 300 (Bilanz und Finanz-Reporting)

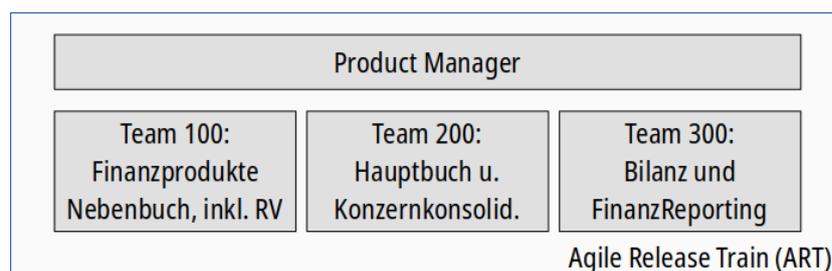


Abb. 1: Vereinfachte Projektstruktur

4.3 PI-Planning

Im Rahmen des ersten Planungsschrittes wurden im PI-Planning folgende Storys mit entsprechenden Abhängigkeiten identifiziert

Scaled Agile: Echtzeitplausibilisiertes PI-Planning

Team#	Story#	Inhalt	SPs	Präzedenz	Ressourcen
alle	N1	PI KickOff	3		A..G
100	N110	Grundlagen/Buchungsstrategie	13	N1	A, B
100	N121	Buchungsregeln plainvanilla Kredite	13	N110	A, B
100	N122	Buchungsregeln Bonds	20	N110	A, B
100	N131	Buchungsregeln Schuldscheindarl.	20	N121,N122	A, B
100	N132	Buchungsregel Eigenemissionen	20	N121,N122	A, B
200	N210	Kontenplan Strategie (GAAP)	13	N1	C, D
200	N220	Gerüst Kontenplan	20	N210, N110	C, D
200	N231	Dimensionen Buchungsbeleg	40	N220, N320	C, D
200	N232	Gerüst Zeitschiene	13	N220	C, D
300	N311	Anforderungen Balance Sheet	20	N1	E, F
300	N312	Anforderungen Financial Reporting	8	N1	E, F
300	N320	Balance Sheet Dimensionen	40	N311,N312	E, F
300	N330	Financial Reporting Dimensionen	13	N320	E, F
alle	N9	Planung des nächsten PI	20	N131,N132, N231,N232, N330	A..G

Tab. 1: Übersicht der Storys im PI mit Komplexitätsschätzung (Storypoints)

Aus den Vorgänger-Nachfolger-Beziehungen der Storys ergibt sich folgender Graph:

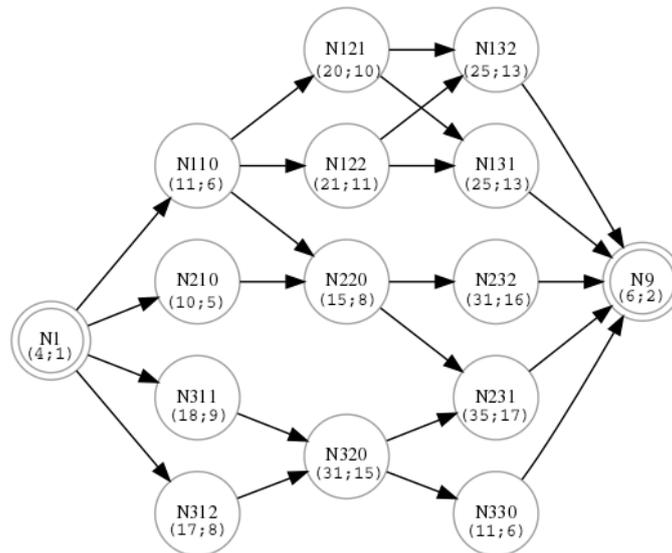


Abb. 2: Präzedenzgraph der Storys im aktuellen PI [inkl. Verteilungsparameter aus 4.4]

Folgende Ressourcenverfügbarkeit (inkl. geplanter Abwesenheiten) wurde festgestellt:

Ressource	Sprint 1	Sprint 2	Sprint 3	Planning u. KickOff	Summe
A	10	20	20	10	50 + 10
B	20	20	20	10	60 + 10
C	20	20	20	10	60 + 10
D	15	15	20	10	50 + 10
E	20	15	20	10	55 + 10
F	20	20	10	10	50 + 10
G	10	... + 10

Tab. 2: Ressourcenverfügbarkeiten PI in Personentagen (PT)

Für die drei Sprints des PI verfügen die Teams 100 und 200 über jeweils 110 PT (\cong 88 SP), Team 300 über 105 PT (\cong 84 SP). Für das Planning&KickOff stehen in Summe 70 PT (\cong 56 SP) zur Verfügung. Dem gegenüber stehen Anforderungen von je 86 SP für Team 100 und 200, 81 SP für Team 300 sowie 23 SP für die Planning- und KickOff-Phase.

Die im SAFe Framework beschriebenen Methoden beschränken sich auf die sprintweise gegebene Plausibilität der Präzedenzen sowie auf die Gegenüberstellung von benötigten und verfügbaren Storypoints. Damit scheint unsere beispielhafte PI-Planung plausibel. Mögliche Konflikte durch Ressourcen-Überlastung, insbesondere im Kontext der Präzedenzen, müssten nun „von Hand“ erkannt werden, was bereits in diesem simplen Beispiel schwierig ist. Nun kommt unser Tool ins Spiel.

4.4 Plausibilisierung der Planung

Zuerst machen wir uns nochmals bewusst, auf welcher grobgranularer Basis (adaptierte Fibonacci-Zahlen aus „Storypoint-Poker“) die Planung erfolgt ist. Eine probabilistische Handhabung dieser Stellgrößen – wie zuvor beschrieben – drängt sich also auf. Hierzu summieren wir die pro Team und Sprint verfügbaren Ressourcen-PT und rechnen diese über die Sprintlaufzeit (20 Arbeitstage) in Mitarbeiterkapazitäten (MAK) um.

Team	Sprint 1	Sprint 2	Sprint 3	Planning KickOff
100	30 PT 1,5 MAK	40 PT 2.0 MAK	40 PT 2.0 MAK	
200	35 PT 1,75 MAK	35 PT 1,75 MAK	40 PT 2.0 MAK	70 PT = 7.0 MAK
300	40 PT 2.0 MAK	35 PT 1,75 MAK	30 PT 1.5 MAK	

Tab. 3: Ressourcenverfügbarkeit pro Team und Sprint in PT und MAK

Die so ermittelten MAK Werte werden nun auf die Storys des jeweiligen Teams bzw. Sprints angewendet, wobei dabei meist nur eine Story zu bearbeiten ist. Anders bei den **fett gedruckten** Zellen, wo jeweils zwei Stories um die verfügbaren Ressourcen konkurrieren, was durch eine SP-anteilige Anrechnung gelöst wird: Die 2.0 MAK aus Team 100, Sprint 2 verteilen sich beispielsweise auf die Stories N121 und N122 im Verhältnis 13:20.

Wie bereits angesprochen, verwenden wir der Einfachheit halber die Normalverteilung, mit den formel-berechneten Parametern μ und σ . Die so ermittelten Parameter (μ, σ) sind in Abb. 2, jeweils direkt unterhalb der Story-Nummer angegeben.

4.5 Ergebnisse

Unser Tool benötigt für die Ergebnisberechnung (auf einem Ryzen-7/3900X Prozessor, mit Schrittweite 0,5) 8 Sekunden. Die ersten beiden Schritte bei der Berechnung erfolgen numerisch, durch drei parallele und eine serielle Reduktion, woraus das nachfolgend abgebildete Zwischen-Konstrukt resultiert. Die aus diesem Reduktionsschritt resultierenden Knoten sind dabei größer und mit 5-stelliger Nummer dargestellt (welche sich aus der Nummer der letzt-erzeugenden Iteration und dem Minimum der beteiligten Originalknotennummern zusammensetzt): so ist z.B. der neue Knoten N20311 durch die zuerst parallele und dann serielle Reduktion der Originalknoten N311, N312 bzw. N320 entstanden.

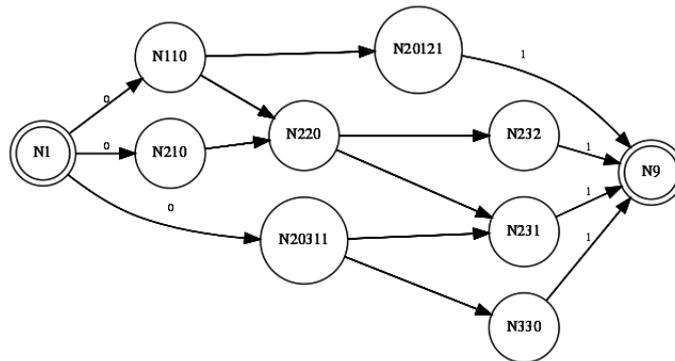
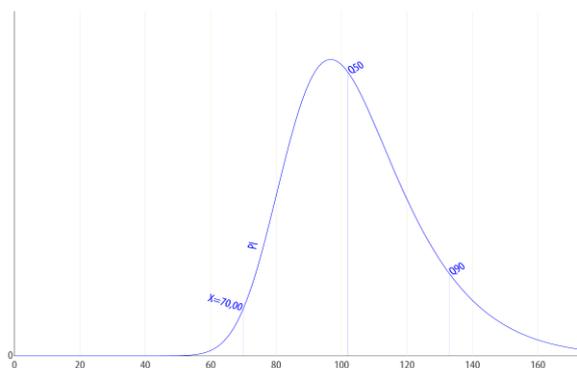


Abb. 3: Präzedenzgraph nach den ersten beiden serienparallelen Reduktionsrunden

Der Teilgraph zwischen N1 und N9 bildet offenbar ein komplexes Cluster, so dass dessen Zielverteilung nun über ein Fitting mit dem PRISM Model Checker zu rechnen ist. Der daraus resultierende Knoten ist schließlich wieder numerisch (seriell) mit N1 und N9 zu reduzieren, was zur Dichte der Gesamtlaufzeit des PI in Arbeitstagen führt:



Zielverteilung PI-Dauer in Arbeitstagen (Soll-Länge = 70 Arbeitstage)

Die definierten Metriken liefern für das so geplante PI folgende Gesamtsicht:

- die Chance, alle Storys plan-gerecht, also binnen 70 Arbeitstagen abzuarbeiten, liegt bei lediglich 1,6%
- bei einem Konfidenz-Niveau von 50% wäre das PI zu 145% überplant
- bei einem Konfidenz-Niveau von 90% wäre das PI zu 189% überplant

Offensichtlich sind die Ziele des PI in Summe also „deutlich zu sportlich“ gewählt. Und das, obwohl die ursprüngliche SAFe-Planung plausibel erschienen war.

5 Schlussfolgerungen und Ausblicke

Unser Tool bietet offenbar ein geeignetes Echtzeit-Instrumentarium, um die Qualität der vernetzten Planung im komplexen agilen Umfeld fundiert bewerten und überarbeiten zu können. Wie das Beispiel zeigt, ist selbst bei gedämpfter Erwartungshaltung (*PI-Ziele werden mit 50% Wahrscheinlichkeit erreicht*) eine Überlastung von knapp 50% zu beobachten. Als Lösung können

- Storys herausgenommen
- die Komplexität bestehender Storys verkleinert
- Bad-Multitasking-Probleme [5] bei konkurrierenden Storys beseitigt
- zusätzliche Ressourcen auf die Storys gesetzt

werden. Die 50%-Chance auf integrale Erfüllung aller DoDs des PI orientiert sich derweil an gängigen Planungspraktiken (z.B. der Critical-Chain-Methode [6]). Sie wird jedoch im fortgeschrittenen Projektverlauf – selbst im agilen Umfeld – möglicherweise Probleme verursachen, denn verbindliche Einsatztermine und/oder fixe Abhängigkeiten mit externen Partnern bedürfen meist einer Vorlaufzeit einer oder mehrerer PI-Längen, so dass die Verwendung der 90%-Konfidenz hier mutmaßlich der bessere Ansatz ist.

Des Weiteren ist unser Tool keineswegs auf den Einsatz im agilen Umfeld beschränkt, denn es können beliebige Präzedenzgraphen mit probabilistischen Zeitdauern der Knoten damit analysiert werden. Dies gilt nicht nur für Vorgangs-Knoten-Netze klassischer Projektsteuerungs-Methoden (z.B. *Critical Path Method* [12], *Critical Chain Method* [6]), sondern prinzipiell auch für das Themenfeld *Prozessmodellierung*, wo für Laufzeitaussagen bislang ebenfalls häufig per Simulation gerechnet wird.

Ein anderes, viel diskutiertes Einsatzgebiet ist die Lösung von Ressourcen-Konflikten, speziell im probabilistischen Umfeld: Wie bereits angesprochen, beinhaltet unser Tool auch effiziente Algorithmen zur Identifizierung und Lösung solcher Konflikte, die auf einer probabilistischen Adaptierung des Algorithmus von *Bell & Han* [3] beruhen. Konkurrierende Ressourcenanforderungen „mit Überbuchung“ werden sicher identifiziert und durch Einfügen neuer Kanten im Hill-Climbing-Verfahren effizient gelöst.

Literaturverzeichnis

- [1] Bauer, Heinz *Wahrscheinlichkeitstheorie*, Walter de Gruyter GmbH & Co. KG 1991.
- [2] Baier, Christel and Katoen, Joost-Pieter *Principles of model checking*, MIT Press, 2008.
- [3] Bell, Colin E. and Han, Jaemin *A New heuristic Solution Method in Resource-Constrained Project Scheduling*, Naval Research Logistics, Vol. 38, pp. 315-331, 1991.
- [4] U.S. Dept. of the Navy *Program Evaluation Research Task, Summary Report, Phase I*, Washington, D.C., Government Printing Office, 1958.
- [5] Goldratt, Eliyahu M. *Theory of Constraints*, The North River Press, 1990.
- [6] Goldratt, Eliyahu M. *Critical Chain*, The North River Press, 1997.
- [7] Klar, Rainer et al. *Messung und Modellierung paralleler und verteilter Rechensysteme*, B.G. Teubner, Stuttgart, 1995.
- [8] Knaster, Richard and Leffingwell, Dean *SAFe 5.0 Distilled*, Pearson Addison-Wesley, 2018.
- [9] Kwiatkowska, Marta, Norman, Gethin and Parker, David *PRISM 4.0: Verification of Probabilistic Real-time Systems*, in Proc. 23rd International Conference on Computer Aided Verification (CAV'11), volume 6806 of LNCS, pp. 585-591, Springer, 2011.
- [10] Leffingwell, Dean, Knaster, Richard, Oren, Inbar and Jemilo, Drew *SAFe 4.5 Reference Guide*, Pearson Addison-Wesley, 2018.
- [11] Vogl, Ulrich and Siegle, Markus *A new approach to predicting reliable project runtimes via probabilistic model checking* in Computer Performance Engineering, 14th European Workshop, EPEW 2017, volume 10497 of LNCS, pp. 117-132, Springer, 2017.
- [12] Shtub, Avraham et al. *Project Management: Processes, Methodologies, and Economics*, Pearson Education Limited, pp. 310-313, 2014.