Dependencies

Definition: Dependency

Progress of one action relies upon the timely output of a previous action, or the presence of some specific thing. Source: Strode & Huff [1]

Impacts of Dependencies

1. Reduced freedom of ordering work item starting priority [2]
2. Increased lead-time of work items waiting for dependencies to resolve
3. Friction between teams who have different priorities and rewards

Reduced Ordering Options – One dependency halves options

Feature A	Feature B	A before B
1	2	Yes
2	1	No

One dependency cuts allowed options in half

Feature start order

Two dependencies cuts allowed start options to 1/6th

Feature A	Feature B	Feature C	A before B	A bf B, B bf C
1	2	3	Yes	Yes
1	3	2	Yes	No
2	1	3	No	No
2	3	1	No	No
3	1	2	Yes	No
3	2	1	No	No

Number of Dependencies	Valid Ordering Options
0	100%
1	50%
2	16.667%
3	4.167%
4	0.833%
5	0.278%

Dependencies narrow the valid options for starting work. This is especially debilitating when planning multiteam delivery of features. Even a single dependency between backlog items halves the allowable start order. By two dependencies only 16% valid options remain, by three only 4% start order options are valid. [2]

Work should be prioritized to minimize cost of delay and maximize value. Dependencies between backlog items or other teams erodes the ability to optimize start order.

Taxonomy of Agile Software Project Dependencies

Source: Strode & Huff [1]

A dependency is created when the progress of one action relies upon the timely output of a previous action, or the presence of some specific thing. Dependencies lead to potential or actual constraints on projects. Potential constraints are those that are currently organised or managed well, causing no problems in the progression of a project. Actual constraints are bottlenecks or points in a project that stakeholders are aware of, but have no immediate means to circumvent.

Knowledge dependency

A knowledge dependency occurs when a form of information is required in order for a project to progress. There are four forms of knowledge dependency:

Requirement - a situation where domain knowledge or a requirement is not known and must be located or identified and this affects project progress

Expertise - a situation where technical or task information is known only by a particular person or group and this affects project progress

Task allocation - a situation where who is doing what, and when, is not known and this affects project progress

Historical - a situation where knowledge about past decisions is needed and this affects project

Task dependency

A task dependency occurs when a task must be completed before another task can proceed and this affects project progress. There are two forms of task dependency:

Activity - a situation where an activity cannot proceed until another activity is complete and this affects project progress

Business process - a situation where an existing business process causes activities to be carried out in a certain order and this affects project progress

Resource dependency

A resource dependency occurs when an object is required for a project to progress. There are two forms of resource dependency:

Entity - a situation where a resource (person, place or thing) is not available and this affects project progress

Technical - a situation where a technical aspect of development affects progress, such as when one software component must interact with another software component, and its presence or absence affects project progress

Increased Lead Time and Risk of Delay: Chance of on-time = 1 in 2^n (where n = dependencies)

Team 1	Team 2	Team 3	Team 4

Team 1	Team 2	Team 3

If multiple teams all need to synchronize there work (have serial dependencies on each other), every dependency doubles the chance of being late due to AT LEAST one of the teams delivering late. In fact, the chance of on-time delivery is 1 chance in 2^n (n = number of dependencies).

For example, if 4 teams had dependencies (left table), there is 1 chance in 16 of on-time delivery (15 chances in 16 of being late due to at least one delay). If one dependency can be removed leaving just 3 dependent teams (right table), the risk of delay is now 1 chance in 8.

Note: a red cell means team delivered late, green means as expected.

References

[1] Strode D, Huff S. A taxonomy of dependencies in agile software development. 23rd Australasian Conference on Information Systems, https://dro.deakin.edu.au/eserv/DU:30049080/strode-taxonomyofdependencies-2012.pdf

[2] Sheerer A., Bick S., Hildenbrand T. and Heinzl A. The Effects of Team Backlog Dependencies on Agile Multiteam Systems: A Graph Theoretical Approach. http://conferences.computer.org/hicss/2015/papers/7367f124.pdf

For more probabilistic forecasting resources: FocusedObjective.com Email: troy.magennis@focusedobjective.com