Online Supplement to

Optimal Software Development: A Control Theoretic Approach

Yonghua Ji [*] • Vijay S. Mookerjee [+] • Suresh P. Sethi [+]

[*] *School of Business, University of Alberta, Edmonton, Alberta T6G 2R6*
[+] *School of Management, University of Texas at Dallas, Richardson, TX 75080*

{yji@ualberta.ca • vijaym@utdallas.edu • sethi@utdalas.edu }

Appendix A

We will encounter the following form of differential equation extensively in this paper:

$$\dot{m}(t) = f(t) \cdot m(t) + q(t) \tag{A.1}$$

The solution to this equation with the initial condition $m(t_1) = m_1$ is:

$$m(t) = e^{\int_0^t f(x)dx} \left[e^{-\int_0^{t_1} f(x)dx} m_1 + \int_{t_1}^{t} q(x) \cdot e^{-\int_0^x f(y)dy} dx \right] \tag{A.2}$$

A. Proof of Theorem 1

We first discuss the general form of the solution and then derive the analytical expression. At the start of the project, the value of u should be 1, i.e., all effort must initially be allocated to development since there is nothing to test at the beginning. Thus we set $u = 1$ until some time. For a small project all the modules are completed and only debugging is needed. Then we have $u = 0$. For a large project, we need to keep constructing while debugging becomes necessary to control the error growth. Then we have $u < 1$, implying that some testing and debugging should begin. There are two sub-cases for a large project we need to consider. First, when the project duration T is large, it's feasible to finish all modules by doing concurrent development and testing. Let T_0 be the time when all the N modules are finished, i.e., $n(T_0) = N$. Beyond T_0, no further development is needed so it is optimal to set $u = 0$ for the rest of the project duration and put all the effort on testing and debugging. We call such projects *relaxed*. The second case is that it is not feasible to finish all N modules using concurrent development and debugging. Then at some time T_2, all the effort is again devoted to write modules. Between T_2 and T, we have $u = 1$. We call such projects *tight*. We solve the problem for a large project and show that the solution form for a small project is an extreme case of the solution for a large relaxed project.

Case 1a: Large Relaxed Project

In the beginning, $u = 1$ for $0 < t \le T_1$ (T_1 is to be determined), we solve (5) analytically to get $n(t)$:

$$n(t) = \left(-k_0 + \sqrt{k_0^{\ 2} + 2k_1 t}\right)/k_1 \qquad \text{(A.3)}$$

Plugging into (A.2) the initial conditions $t_1 = 0$ and $m_1 = 0$ together with the expressions $f(t) = k_3 \dot{n}(t)$ and $q(t) = [k_2 + k_4 n(t)]\dot{n}(t)$ obtained from (6), we have:

$$m(t) = -k_4 n / k_3 + (k_4 + k_2 k_3)/k_3^2 \cdot \left(e^{k_3 n} - 1\right) \qquad \text{(A.4)}$$

Next, in the interval $T_1 < t < T_0$ (T_0 is also to be determined), $0 < u(t) < 1$. Then from Equation (11) we have $H_u = 0$ in order for $u(t)$ to be optimal. In order for H_u to stay zero in this interval, we should have $\dot{H}_u = 0$. From Equation (11), we can see H_u is a function of variables *n*, *m*, λ_1 and λ_2. Then $\dot{H}_u$ is a function of $\dot{n}$, $\dot{m}$, $\dot{\lambda}_1$ and $\dot{\lambda}_2$. Plugging the expressions of (1), (2), (9) and (10) into $\dot{H}_u$ and after some algebraic manipulation, we can show that

$$\dot{H}_u = -\frac{\lambda_2 k_5}{n^2 (k_0 + k_1 n)}(m - k_2 n - k_4 n^2) \qquad \text{(A.5)}$$

So the requirement of $\dot{H}_u = 0$ leads to the following equation:

$$m(t) = [k_2 + k_4 n(t)]n(t) \qquad \text{(A.6)}$$

Plugging this equation into (6), we can obtain:

$$\dot{n}(t) = \frac{-k_5 m / n}{k_4 n - k_3 m - (k_0 + k_1 n) \cdot k_5 m / n} \qquad \text{(A.7)}$$

Then from (5), we have

$$u(t) = \frac{-(k_0 + k_1 n) \cdot k_5 m / n}{k_4 n - k_3 m - (k_0 + k_1 n) \cdot k_5 m / n} \qquad \text{(A.8)}$$

Construction effort $u(t)$ in the interval $T_1 < t < T_0$ can be shown to decrease over time *t*. Therefore, the debugging effort will increase as the project continues in this time interval. This result is intuitive: as time moves on, more attention should be devoted to debugging to maintain system quality.

In our problem, the Hamiltonian (8) is linear in the control variable *u*. An additional necessary condition analogous to the second-order condition in calculus, called *the generalized Legendre-Clebsch Condition*, $(-1)\frac{\partial}{\partial u}\left[\frac{d^2 H_u}{dt^2}\right] < 0$ is verified to hold in this problem.

In the third interval of $T_0 \le t \le T$, $u(t) = 0$ and $n(t) = N$ since all *N* modules are finished and all the programming effort is spent on testing modules to remove inter-modules errors. Solving equation (6), we can get $m(t)$:

$$m(t) = m(T_0) e^{-k_5 (t - T_0)/N} \tag{A.9}$$

The number of errors $m(t)$ decreases exponentially as a function of time t. This is exactly the form of the Goel-Okumoto software reliability growth model (Goel and Okumoto, 1979) which is used extensively in the software reliability literature. The Goel-Okumoto model is a special case of our more general model where $u(t) = 0$.

To completely solve the problem, we need to find the values of two switching points T_1 and T_0. We can solve $N_0 (= n(T_1))$ by equating equations (A.4) and (A.6) at T_1 since $m(t)$ is a continuous function of time *t*. N_0 can only be obtained numerically. Then from (A.3), we can obtain T_1 as the following:

$$T_1 = \left(k_1 N_0^2 + 2k_0 N_0\right)/2 \tag{A.10}$$

The threshold N_0 is used to classify projects between large and small ones. If the number of modules to be finished in a project *N* is larger than N_0, then some additional construction effort is required beyond T_1. Such a project is termed *large*. If *N* is smaller than N_0, then the full construction effort period stops at some time $T_1^{'}$ (earlier than T_1) and full debugging effort takes over for the rest of the project duration. Such a project is termed *small*. The solution form of a small project is an extreme case of the solution form of a large relaxed project. The value of $T_1^{'}$ is obtained from (A.3) as shown in the following:

$$T_1^{'} = \left(k_1 N^2 + 2k_0 N\right)/2 \tag{A.10a}$$

To find the switching point T_0, we solve the differential equation (A.7):

$$t = \frac{k_1 n(t)^2 + 2k_0 n(t)}{2} + \frac{k_3 [n(t)^2 - N_0^2]}{2k_5} - \frac{n(t) - N_0}{k_5} + \frac{k_2}{k_4 k_5} Log\left(\frac{k_2 + k_4 n(t)}{k_2 + k_4 N_0}\right) \tag{A.11}$$

At T_0, all N modules are finished, i.e., $n(T_0)=N$. So from (A.11), we determine T_0 as the following:

$$T_0 = \frac{k_1 N^2 + 2k_0 N}{2} + \frac{k_3(N^2 - N_0^2)}{2k_5} - \frac{(N-N_0)}{k_5} + \frac{k_2}{k_4 k_5} Log\left(\frac{k_2 + k_4 N}{k_2 + k_4 N_0}\right). \tag{A.12}$$

Case 1b: Large Tight Project

The solutions in the three time intervals are very similar to those in Case 1a except in the third interval where $T_2 < t \le T$. The switching point T_1 is the intersection of the curves (A.4) and (A.6) which is same for both cases. Thus we obtain the same value of T_1 for both cases. Hence we focus on finding the switching point T_2.

In the interval $T_2 \le t \le T$, $u(t)=1$. We can solve Equation (5) analytically to get $n(t)$:

$$n(t) = \frac{-k_0 + \sqrt{k_0{}^2 + 2k_1 t'}}{k_1} \tag{A.13}$$

where $t' = t - \left(T - \frac{k_1 N^2 + 2k_0 N}{2}\right)$. By utilizing the general form of *m(t)* in (A.2), we solve $m(t)$ as:

$$m(t) = -\frac{k_4 n}{k_3} + \frac{k_4 + k_2 k_3}{k_3^2}\left(e^{k_3(n-N_2)} - 1\right) + e^{k_3(n-N_2)} \frac{k_3 M_2 + k_4 N_2}{k_3} \tag{A.14}$$

where $N_2 = n(T_2)$ and $M_2 = m(T_2)$.

The second switching time T_2 needs to be solved by equating two equations. On one hand, $n(t)$ satisfies the same module growth equation (A.7) in the interval $T_1 < t \le T_2$, as in Case 1a and thus the same relationship holds between time t and $n(t)$ as described by (A.11). We can get T_2 from (A.11):

$$T_2 = \frac{k_1 N_2^2 + 2k_0 N_2}{2} + \frac{k_3[N_2^2 - N_0^2]}{2k_5} - \frac{N_2 - N_0}{k_5} + \frac{k_2}{k_4 k_5} Log\left(\frac{k_2 + k_4 N_2}{k_2 + k_4 N_0}\right) \tag{A.15}$$

On the other hand, by evaluating $n(t)$ at T_2, we can get the following equation from (A.13):

$$T_2 = \left(T - \frac{k_1 N^2 + 2k_0 N}{2}\right) + \frac{k_1 N_2^2 + 2k_0 N_2}{2} \tag{A.16}$$

We can solve for N_2 by equating the two equations (A.15) and (A.16). Then we can plug N_2 back into either (A.15) or (A.16) to get T_2.

B. Expressions for λ_1 and λ_2

We point out that λ_1 is positive and λ_2 is negative during the whole project duration. To see this, we can simplify (10) to be $\dot{\lambda}_2 = -\lambda_2 \left[k_3 \dot{n} - k_5 (1-u)/n \right]$, the solution to which is:

$$\lambda_2(t) = \lambda_2(T) e^{\int_t^T [k_3 \dot{n} - k_5 (1-u)/n] dx} \tag{A.17}$$

The terminal condition is $\lambda_2(T) = -1$; our objective is to minimize $m(T)$. Therefore, λ_2 is negative during the whole project duration for both large and small projects.

We verify that λ_1 is positive for a large project. From Equation (11), we define $h_1 = (k_0 + k_1 \cdot n)^{-1}$ and $h_2 = [c_1(n,m)/(k_0 + k_1 \cdot n) + c_2(n,m)]$. Then in the interval $0 < t < T_1$, we have

$$\lambda_1 = (H_u - \lambda_2 h_2)/h_1 \tag{A.18}$$

where H_u can be obtained easily by integrating (A.5) with terminal condition $H_u(T_1) = 0$:

$$H_u(t) = \int_t^{T_1} \frac{\lambda_2 k_5}{n^2 (k_0 + k_1 n)} (m - k_2 n - k_4 n^2) \cdot dx \tag{A.19}$$

We can see that $H_u(t)$ is positive in this interval since $m < k_2 n + k_4 n^2$ in this interval. Therefore λ_1 is positive in this interval since λ_2 has been shown to be negative in (A.17) and h_2 is always positive.

In the interval $T_1 < t < T_0$ for a relaxed project and $T_1 < t < T_2$ for a tight project, $H_u(t) = 0$. Hence,

$$\lambda_1 = -\lambda_2 h_2 / h_1 > 0 \tag{A.20}$$

Finally, in the interval $T_0 < t < T$ for a relaxed project, we have $u = 0$. Then from (9), we have $\dot{\lambda}_1 = -\lambda_2 k_5 m / n^2 > 0$. Then,

$$\lambda_1(t) = \lambda_1(T_0) - \int_{T_0}^{t} \lambda_2 k_5 m / n^2 dx > 0 \tag{A.21}$$

For a tight project in the interval $T_2 < t < T$, we have the same expression for λ_1 as given in (A.18) except that H_u is now given by:

$$H_u(t) = -\int_{T_2}^{t} \frac{\lambda_2 k_5}{n^2 (k_0 + k_1 n)} (m - k_2 n - k_4 n^2) \cdot dx \tag{A.22}$$

with the initial condition $H_u(T_2) = 0$. Notice that $m > k_2 n + k_4 n^2$ in this interval. Therefore we have positive $H_u(t)$ and hence positive λ_1 in this interval.

Again, a small project is just an extreme case of a large relaxed project. The value of λ_1 is given by (A.18) for $0 < t < T_1^{'}$ and by (A.21) for $T_1^{'} < t < T$ with $T_1^{'}$ replacing T_1 in (A.19) and T_0 in (A.21).

C. Proof of Corollary 1 by Contradiction

Suppose we have a situation in Figure 5 where *u*=0 for $T_i < t < T_{i+1}$ and *u*=1 for $T_{i+1} < t < T_{i+2}$. At the switching point T_i, we have H_u =0 according to the necessary condition (11) since H_u is continuous in this problem. According to (A.5), we have

$$H_u(t) = -\int_{T_i}^{t} \frac{\lambda_2 k_5}{n^2(k_0 + k_1 n)}(m - k_2 n - k_4 n^2)\cdot dx \tag{A.23}$$

which remains negative for $T_i < t < T_{i+1}$ as long as it becomes negative after T_i. The reason is that, when all effort is spent on debugging (*u*=0), *m* is always decreasing and *n* stays the same. In addition, the adjoint variable λ_2 is shown in Appendix B to be negative during the whole project duration. Since H_u is negative at T_{i+1}, due to continuity, H_u is also negative at T_{i+1}^{+}. Thus *u*=1 for $T_{i+1} < t < T_{i+2}$ would violate the necessary condition (11). Therefore, alternating between pure construction and pure debugging cannot be optimal.

D. Signs of dN_0 / dk_i, *i*=2,3,4

The threshold value N_0 is the positive intersection point of two curves (A.4) and (A.6), i.e, it is the positive solution of $s(n, k_2, k_3, k_4) = 0$ where

$$s(n, k_2, k_3, k_4) = -\frac{k_4 n}{k_3} + \frac{k_4 + k_2 k_3}{k_3^2}\left(e^{k_3 n} - 1\right) - \left(k_2 + k_4 n\right)n, \tag{A.24}$$

Then we have

$$\frac{dN_0}{dk_i} = -\frac{\partial s(n, k_2, k_3, k_4)/\partial k_i}{\partial s(n, k_2, k_3, k_4)/\partial n}\Big|_{n=N_0}, \; i = 2, 3, 4 \tag{A.25}$$

We first show that $\frac{\partial s}{\partial n}\big|_{n=N_0}$ is positive. First we have $s(0, k_i) = 0$, $\frac{\partial s}{\partial n}\big|_{n=0} = 0$ and $\frac{\partial^2 s}{\partial n^2}\big|_{n=0} = k_2 k_3 - k_4$. We require $k_2 k_3 - k_4 < 0$ in order to have a positive N_0 since $\frac{\partial^2 s}{\partial n^2}$ increases monotonically ($\frac{\partial^3 s}{\partial n^3} = k_3(k_4 + k_2 k_3)e^{k_3 n} > 0$ for all n)[1]. Therefore, $s(n, k_2, k_3, k_4) = 0$ has only one positive solution. Then $\frac{\partial s}{\partial n}|_{n=N_0} > 0$ since $s(n, k_2, k_3, k_4)$ turns from negative to positive at $n = N_0$. So, the sign of $\frac{dN_0}{dk_i}$ is opposite to the sign of $\frac{\partial s}{\partial k_i}|_{n=N_0}$. And we

$$\frac{\partial s}{\partial k_2}|_{n=N_0} = \frac{1}{k_3}\left(e^{k_3 N_0} - 1\right) - N_0 = \frac{k_3 k_4}{k_4 + k_2 k_3} N_0^2 > 0 \quad \text{(A.26)}$$

$$\frac{\partial s}{\partial k_3}|_{n=N_0} = \frac{N_0^2 k_2 k_3^2 (k_2 + k_4 N_0) + k_4 N_0 \cdot \partial s(n, k_i)/\partial n|_{n=N_0}}{k_3(k_4 + k_2 k_3)} > 0 \quad \text{(A.27)}$$

$$\frac{\partial s}{\partial k_4}|_{n=N_0} = -\frac{N_0}{k_3} + \frac{1}{k_3^2}\left(e^{k_3 N_0} - 1\right) - N_0^2 = -\frac{k_2 k_3}{k_4 + k_2 k_3} N_0^2 < 0 \quad \text{(A.28)}$$

To summarize, we have shown $\frac{dN_0}{dk_2} < 0$, $\frac{dN_0}{dk_3} < 0$, and $\frac{dN_0}{dk_4} > 0$.

When $k_4 N_0$ is much greater than k_2 and k_4 much greater than $k_2 k_3$, then (A.24), which N_0 should satisfy, becomes

$$0 = -n/k_3 + \left(e^{k_3 n} - 1\right)/k_3^2 - n^2.$$

Therefore N_0 becomes independent of k_4 under both conditions.

E. Signs of dT_0/dk_i, *i*=0,…,5

Similarly, we can perform a sensitivity analysis on $T_0(N_0, k_0, ..., k_5)$ with respect to k_i, i=0,…, 5. From Equation (A.12), we have,

$$dT_0/dk_0 = N > 0 \quad \text{(A.29)}$$

$$dT_0/dk_1 = N^2/2 > 0 \quad \text{(A.30)}$$

To find the sign of $\frac{dT_0}{dk_2}$, first we have $\frac{\partial T_0}{\partial N_0} = -\frac{1}{k_5(k_2 + k_4 N_0)} \frac{\partial s}{\partial n}|_{n=N_0} < 0$ from (A.12). Also,

---

[1] Considering the case $k_2 k_3 - k_4 > 0$ does not bring more insight except that concurrent construction and debugging starts at time t=0, instead of the point in time after N_0 modules have been constructed.

$$\frac{\partial T_0}{\partial k_2} = \frac{1}{k_4 k_5}\left[Log\left(\frac{k_2 + k_4 N}{k_2 + k_4 N_0}\right) + \frac{k_2}{k_2 + k_4 N} - \frac{k_2}{k_2 + k_4 N_0}\right]$$

which is positive since $\frac{\partial T_0}{\partial k_2}|_{N=N_0} = 0$ and $\frac{d}{dN}\left(\frac{\partial T_0}{\partial k_2}\right) = \frac{1}{k_5}\frac{k_4 N}{(k_2 + k_4 N)^2} > 0$. Then,

$$\frac{dT_0}{dk_2} = \frac{\partial T_0}{\partial k_2} + \frac{\partial T_0}{\partial N_0}\frac{dN_0}{dk_2} > 0 \tag{A.31}$$

$$\frac{dT_0}{dk_3} = \frac{1}{2k_5}\left(N^2 - N_0^2\right) + \frac{\partial T_0}{\partial N_0}\frac{dN_0}{dk_3} > 0 \tag{A.32}$$

It is easy to verify that $\frac{\partial T_0}{\partial k_4} = -\frac{k_2}{k_4}\frac{\partial T_0}{\partial k_2}$. Then we have

$$\frac{dT_0}{dk_4} = \frac{\partial T_0}{\partial k_4} + \frac{\partial T_0}{\partial N_0}\frac{dN_0}{dk_4} < 0 \tag{A.33}$$

The first term in (A.12) is the time needed to finish the whole project by using all the programming effort. Therefore, it is less than T_0. So,

$$\frac{dT_0}{dk_5} = -\frac{1}{k_5}\left(T_0 - \frac{k_1 N^2 + 2k_0 N}{2}\right) < 0 \tag{A.34}$$

Under the conditions that $k_4 N_0$ is much greater than k_2 and k_4 much greater than $k_2 k_3$, we can easily show that T_0 will be independent of k_4 by solving Equation (A.7) directly:

$$T_0 = \frac{k_1 N^2 + 2k_0 N}{2} + \frac{k_3(N^2 - N_0^2)}{2k_5} - \frac{(N - N_0)}{k_5}$$

The value of T_0 is independent of k_4 since N_0 has been shown to be independent of k_4 under the same conditions. We can explain the result intuitively. When $k_4 N$ is much greater than k_2, which is true within $T_1 < t < T_0$ in the parameter space we use, from (A.8), $u(t)$ becomes $u(t) = \frac{-(k_0 + k_1 n)\cdot k_5}{1 - k_3 n - (k_0 + k_1 n)\cdot k_5}$, which is independent of k_4. Then T_0 is independent of k_4 since both T_1 and the time to finish the remaining module after T_1 are independent of k_4.

F. Number of bugs *m*(*T*) for the fixed allocation policy

With $u(t) = u_0$, we have $f(t) = k_3\dot{n}(t) - k_5 / n(t)\cdot(1 - u_0)$ and $q(t) = [k_2 + k_4 n(t)]\dot{n}(t)$ from Equation (6). Then we have

$$\int_x^t f(x)dx = \int_x^t \left[k_3 \dot{n}(x) - \frac{k_5}{n(x)}(1-u_0) \right] dx = \left[a \cdot n(t) - b \cdot \ln n(t) \right] - \left[a \cdot n(x) - b \cdot \ln n(x) \right]$$

where $a = k_3 - \frac{k_1 k_5}{u_0}(1-u_0)$ and $b = \frac{k_0 k_5}{u_0}(1-u_0)$.

From (A.2) with initial conditions $t_1 = 0$ and $m_1 = 0$, we have:

$$m(T) = \int_0^T [k_2 + k_4 n(x)] \dot{n}(x) e^{\int_x^T f(y)dy} dx = e^{a \cdot N} N^{-b} \int_0^T [k_2 + k_4 n(x)] \dot{n}(x) e^{-a \cdot n(x)} n(x)^b dx$$

$$= e^{a \cdot N} N^{-b} a^{-(b+1)} \left[k_2 \Gamma_{aN}(b+1) + \frac{k_4}{a} \Gamma_{aN}(b+2) \right] \quad \text{(A.35)}$$

where the incomplete gamma function $\Gamma_z(p)$ is defined in (Korn and Korn, 2000):

$$\Gamma_z(p) = \int_0^Z t^{p-1} e^{-t} dt, \qquad p > 0.$$

In the special case of $u_0 = 1$, it is easy to verify that (A.35) becomes (A.4).

Appendix G: Derivation of the Necessary Condition of Optimal *T*

In order for T to be the optimal software project schedule, we have the first-order terminal condition:

$$H(T) - \dot{G}(T) = 0 \quad \text{(A.36)}$$

Also, $\lambda_2(t)$ needs to satisfy the terminal condition:

$$\lambda_2(T) = -h'(m(T)).$$

We can rewrite $H(t) = H_u u(t) - \lambda_2 k_5 m(t) / n(t)$. Then (A.36) can be written as:

$$\dot{G}(T) = H_u(T) u(T) + k_5 m(T) / N \cdot h'(m(T)) \quad \text{(A.37)}$$

Only for large tight projects (Figure 4) can we have $u(T) \neq 0$. We only need to find an expression for $H_u(T)$ for such projects. Using the expression for $\dot{H}_u$ in (A.5), we can find $H_u(T)$ for large tight projects as

$$H_u(T) = -\int_{T_2}^T \frac{\lambda_2 k_5}{n^2 (k_0 + k_1 n)} (m - k_2 n - k_4 n^2) \cdot dt \quad \text{(A.38)}$$

where $\lambda_2(t)$ is given in (A.17).