

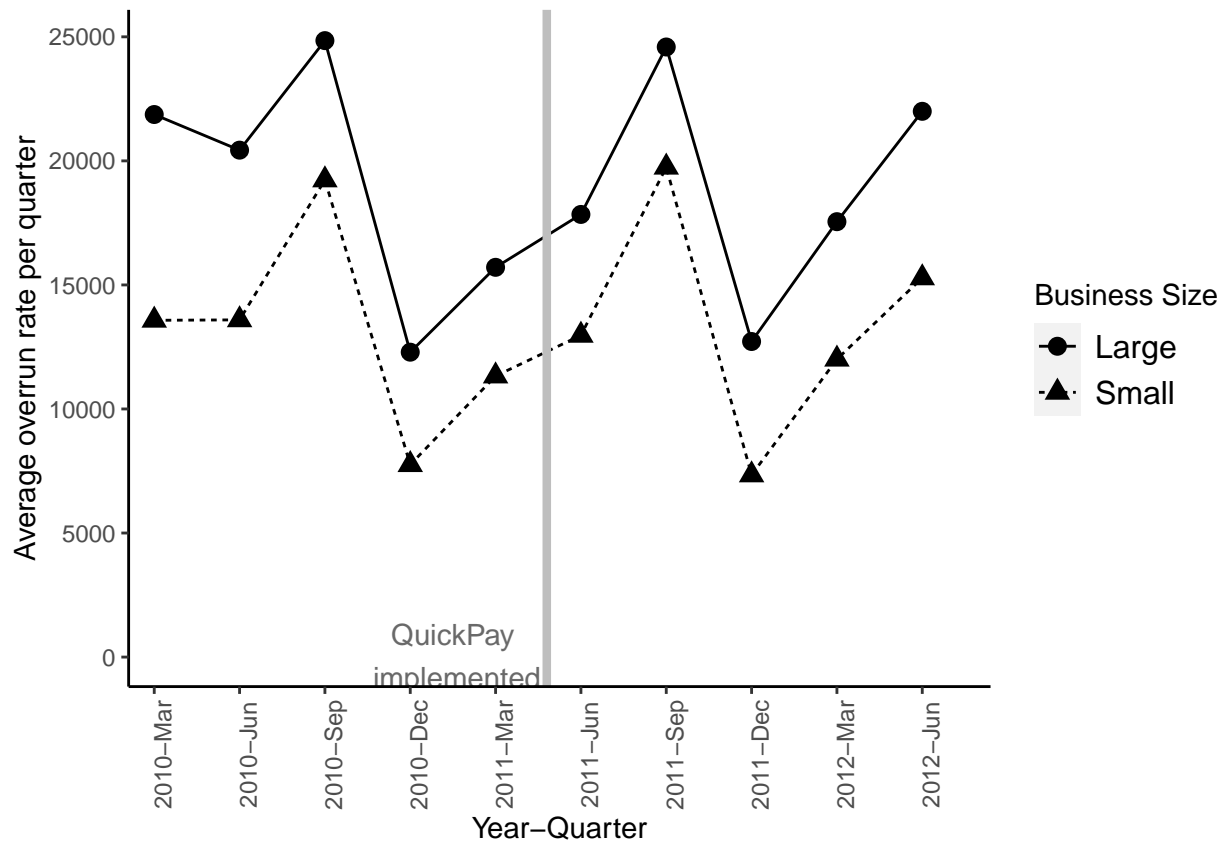
Budget Overruns: First Implementation of QuickPay (2009-2012)

Sep 19, 2021

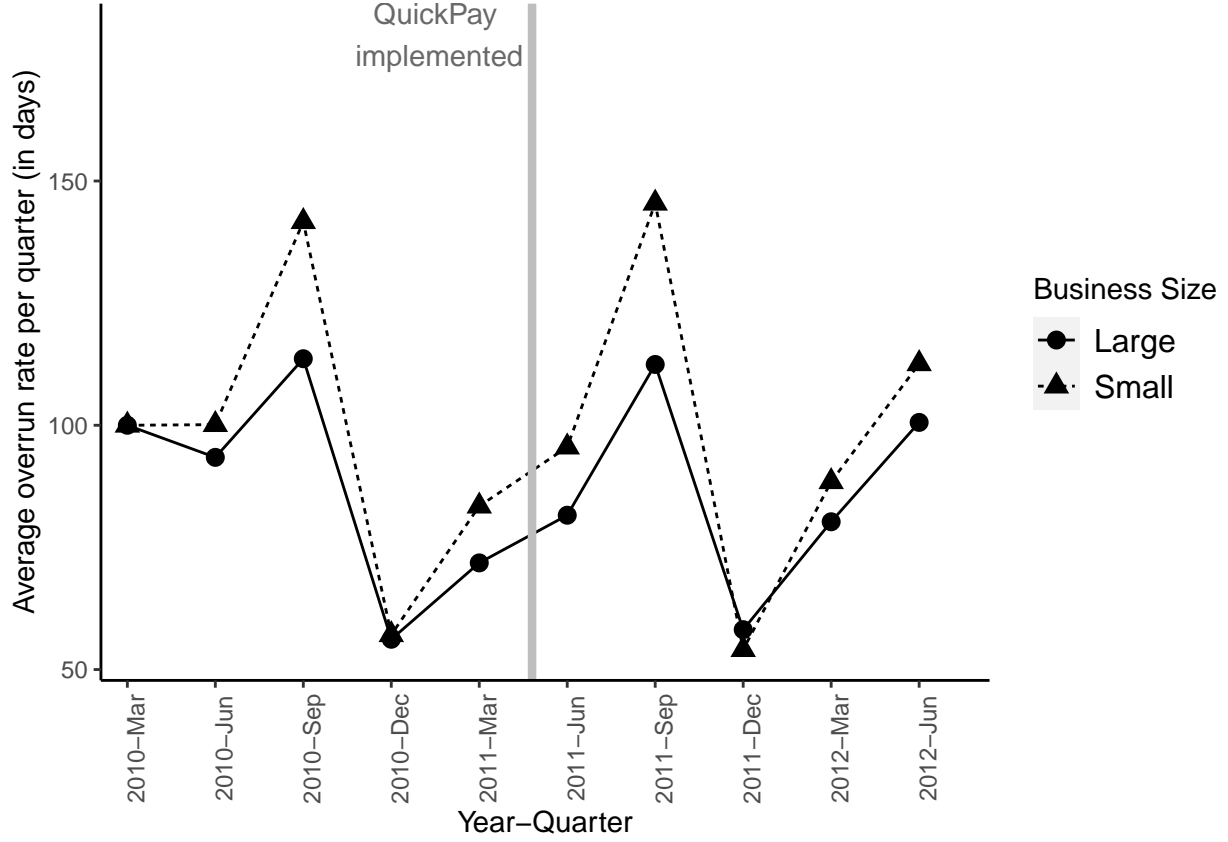
1 Note

- Below is the definition of `base_and_all_options_value` from the data dictionary:
 - The change (from this transaction only) to the potential contract value (i.e., the base contract and any exercised or unexercised options).
- This means that every observation in raw data shows incremental change from previous budget. So some of the values can be zero.
- We, therefore, need to calculate the new budget at each point in time (by adding all previous values). We did this in the resampling step, but mentioning here for reference.
- This is different from calculation of delays, where `period_of_performance_current_end_date` indicated the new deadline of the project.

2 Budget Overrun over Time



2.1 Normalized Overrun



3 Notation

- Project i , Year-Quarter t
- X_i denotes project level controls: initial duration, initial budget, number of offers received
- $\mu_t, \theta_{firm}, \lambda_{task}$: Year-Quarter, Firm, and Product/Service code Fixed effects
- All continuous variables are winsorized at the 5% level

$$Treat_i = \begin{cases} 1, & \text{if project } i \text{ is a small business} \\ 0, & \text{otherwise} \end{cases}$$

$$Post_t = \begin{cases} 1, & \text{if year-quarter } t > \text{April 27, 2011} \\ 0, & \text{otherwise} \end{cases}$$

4 Baseline Regressions

$$Overrun_{it} = \alpha + \beta_0 Treat_i + \beta_1 Post_t + \beta_2 (Treat_i \times Post_t) + \epsilon_{it}$$

$$\begin{aligned} Overrun_{it} = & \alpha + \beta_0 Treat_i + \beta_1 Post_t + \beta_2 (Treat_i \times Post_t) \\ & + X_i + (Post_t \times X_i) + \mu_t + \theta_{firm} + \lambda_{task} + \epsilon_{it} \end{aligned}$$

Table 1: Quickpay 2009-2011

| | <i>Overrun_{it}</i> (in days) | | | | |
|--|---------------------------------------|--------------------------|--------------------------|--------------------------|----------------------------|
| | (1) | (2) | (3) | (4) | (5) |
| <i>Treat_i</i> | -5,606.99*** (442.41) | -4,860.95*** (467.78) | -4,580.56*** (463.49) | -4,247.65*** (485.36) | -4,727.72*** (1,267.62) |
| <i>Post_t</i> | 600.14 (373.36) | -5,515.13*** (479.98) | | | |
| <i>Treat_i × Post_t</i> | 114.97 (484.85) | 841.83 (512.87) | 685.19 (510.36) | 843.73* (507.59) | 404.57 (532.11) |
| Constant | 18,064.32*** (348.01) | 6,930.79*** (439.67) | | | |
| Duration, Budget, Bids | No | Yes | Yes | Yes | Yes |
| <i>Post_t × (Duration, Budget, Bids)</i> | No | Yes | Yes | Yes | Yes |
| Project Age Tercile | No | Yes | Yes | Yes | Yes |
| Year-Quarter Fixed Effects | No | No | Yes | Yes | Yes |
| Task Fixed Effects | No | No | No | Yes | Yes |
| Firm Fixed Effects | No | No | No | No | Yes |
| Observations | 189,371 | 168,851 | 168,851 | 168,851 | 168,851 |
| R ² | 0.003 | 0.03 | 0.04 | 0.09 | 0.23 |
| Adjusted R ² | 0.003 | 0.03 | 0.04 | 0.08 | 0.17 |

Note:

*p<0.1; **p<0.05; ***p<0.01

Each observation is a project-quarter.

SEs are robust and clustered at the project level.

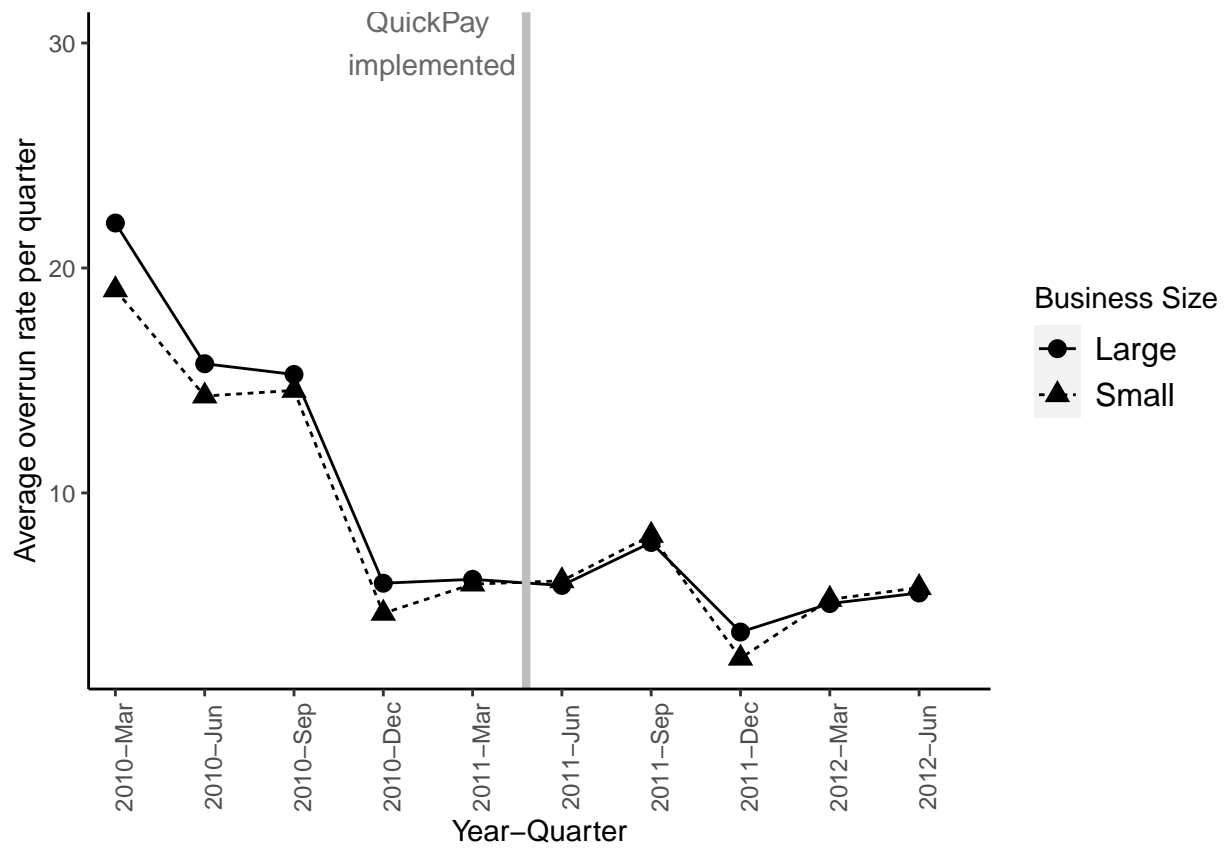
5 Percentage Overrun

$$PercentOverrun_{it} = \beta_0 + \beta_1 Treat_i + \beta_2 Post_t + \beta_3 (Treat_i \times Post_t) + e_{it}$$

$$\begin{aligned}
 PercentOverrun_{it} = & \alpha + \beta_0 Treat_i + \beta_1 Post_t + \beta_2 (Treat_i \times Post_t) \\
 & + X_i + (Post_t \times X_i) + \mu_t + \theta_{firm} + \lambda_{task} + \epsilon_{it}
 \end{aligned}$$

5.1 Percentage Overrun over time

- Sample restricted to projects with modification zero when they first appeared in our sample.
- $PercentOverrun_{it} = 100 \times Overrun_{it} / Budget_{i,t-1}$



5.1.1.1 Normalized Overrun

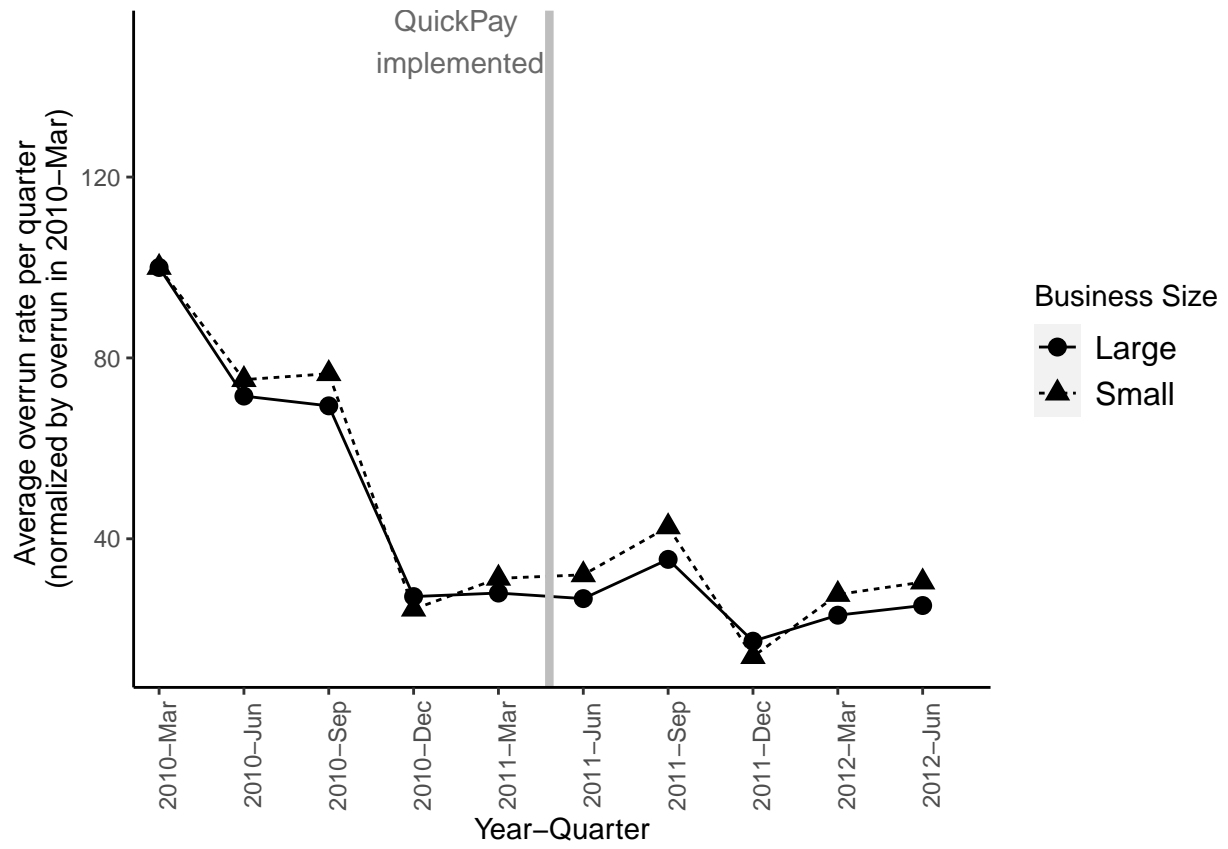


Table 2: Effect of QuickPay on project overrun rates

| | <i>PercentOverrun_{it}</i> | | | | |
|--|------------------------------------|--------------------|--------------------|--------------------|-------------------|
| | (1) | (2) | (3) | (4) | (5) |
| <i>Treat_i</i> | -1.26*** (0.25) | -1.59*** (0.25) | -1.32*** (0.25) | -0.94*** (0.25) | -0.72 (0.65) |
| <i>Post_t</i> | -5.23*** (0.19) | -6.24*** (0.31) | | | |
| <i>Treat_i × Post_t</i> | 1.13*** (0.27) | 1.18*** (0.28) | 0.96*** (0.28) | 1.05*** (0.28) | 1.00*** (0.29) |
| Constant | 10.79*** (0.18) | 14.22*** (0.29) | | | |
| Duration, Budget, Bids | No | Yes | Yes | Yes | Yes |
| <i>Post_t × (Duration, Budget, Bids)</i> | No | Yes | Yes | Yes | Yes |
| Project age | No | Yes | Yes | Yes | Yes |
| Year-Quarter fixed effects | No | No | Yes | Yes | Yes |
| Task fixed effects | No | No | No | Yes | Yes |
| Contractor fixed effects | No | No | No | No | Yes |
| Observations | 160,810 | 150,245 | 150,245 | 150,245 | 150,245 |
| R ² | 0.01 | 0.03 | 0.05 | 0.08 | 0.19 |
| Adjusted R ² | 0.01 | 0.03 | 0.05 | 0.07 | 0.13 |

Note:

*p<0.1; **p<0.05; ***p<0.01

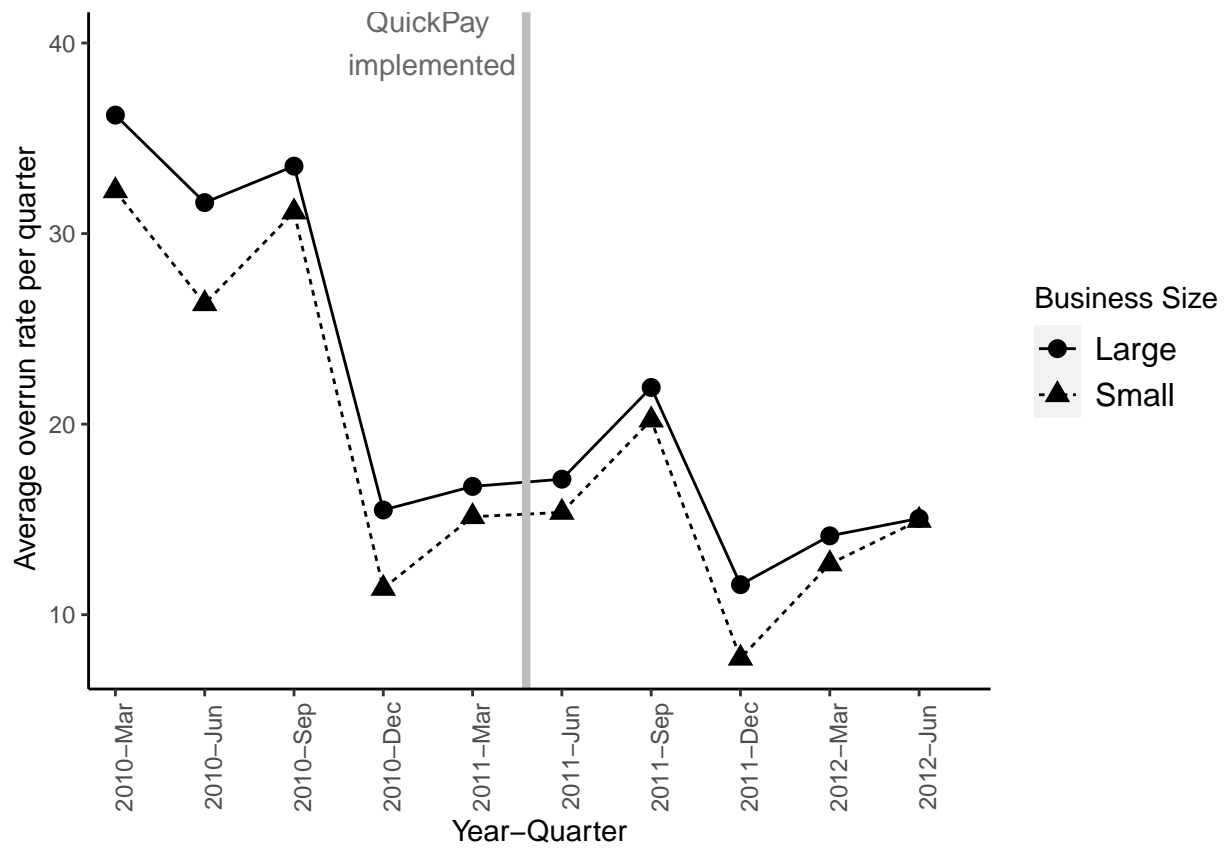
Each observation is a project-quarter.

SEs are robust and clustered at the project level.

6 Relative Overrun

6.1 Relative overruns over time

- Sample restricted to projects with modification zero when they first appeared in our sample.
- $RelativeOverrun_{it} = RelativeOverrun_{it} / InitialDuration_i$



6.1.1 Normalized overrun

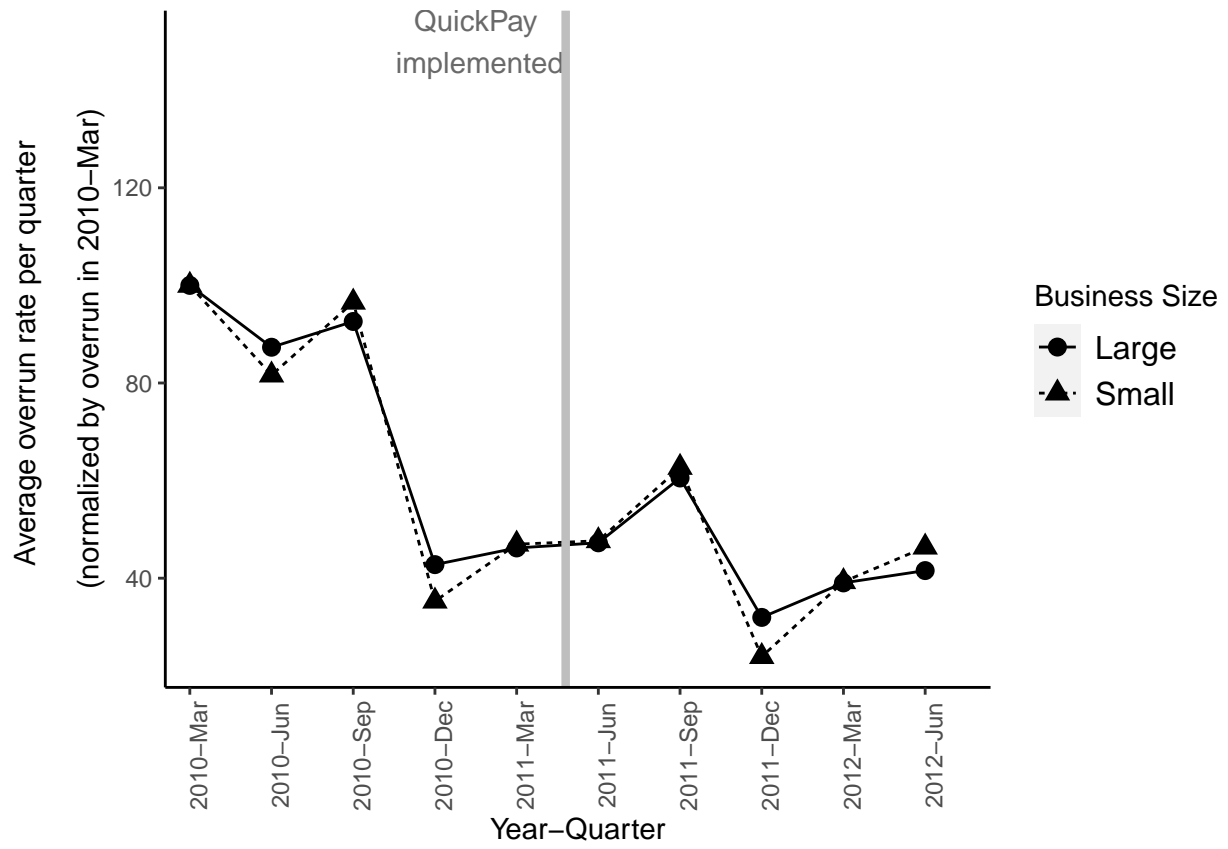


Table 3: Effect of QuickPay on project overrun rates

| | <i>RelativeOverrun_{it}</i> | | | | |
|---|-------------------------------------|--------------------|--------------------|--------------------|-----------------|
| | (1) | (2) | (3) | (4) | (5) |
| <i>Treat_i</i> | -3.63*** (0.67) | -3.96*** (0.65) | -3.61*** (0.65) | -2.26*** (0.66) | -2.47 (1.63) |
| <i>Post_t</i> | -8.40*** (0.45) | -7.83*** (0.57) | | | |
| <i>Treat_i × Post_t</i> | 1.66*** (0.64) | 1.13* (0.63) | 0.87 (0.63) | 0.96 (0.63) | 1.04 (0.66) |
| Constant | 24.28*** (0.49) | 28.57*** (0.57) | | | |
| Budget, Bids | No | Yes | Yes | Yes | Yes |
| <i>Post_t × (Budget, Bids)</i> | No | Yes | Yes | Yes | Yes |
| Project age | No | Yes | Yes | Yes | Yes |
| Year-Quarter fixed effects | No | No | Yes | Yes | Yes |
| Task fixed effects | No | No | No | Yes | Yes |
| Contractor fixed effects | No | No | No | No | Yes |
| Observations | 189,365 | 189,164 | 189,164 | 189,164 | 189,164 |
| R ² | 0.004 | 0.03 | 0.03 | 0.08 | 0.21 |
| Adjusted R ² | 0.004 | 0.03 | 0.03 | 0.07 | 0.16 |

Note:

*p<0.1; **p<0.05; ***p<0.01

Each observation is a project-quarter.

SEs are robust and clustered at the project level.