A. We need the period to be even, so the middle example is useless. Let's try the last period, where $k = 12$ and $p = 4$. We compute

$$k^{p/2} \equiv 12^2 \equiv 144 \equiv 14 \not\equiv -1 \pmod{65}.$$

So this example is useful. We have two options, each of which produces a non-trivial factor of $m = 65$:

$$\gcd(m, k^{p/2} - 1) = \gcd(65, 14 - 1) = 13,$$

and

$$\gcd(m, k^{p/2} + 1) = \gcd(65, 14 + 1) = 5.$$

B.A. The three circuit diagrams are identical, except for the choice of ℓ (the number of repetitions of $(R \otimes I) \cdot F$).

B.B. In the unknown $k \geq 1$ version, we choose ℓ uniformly randomly between 1 and what it would be if we knew that $k = 1$, which is

$$\ell = \text{round}\left(\frac{\pi}{4t} - \frac{1}{2}\right),$$

where $t = \arcsin(2^{-n/2})$.

B.C. We can handle the unknown $k \geq 0$ case by invoking the unknown $k \geq 1$ algorithm repeatedly. On each invocation, we of course check whether the δ produced really satisfies $f(\delta) = 1$. Each invocation has probability about $1/2$ of producing a good δ. If there is a good δ, then we probably find it within the first few invocations, and we know that $k \geq 1$. If we don't find a good δ in m invocations, then the probability that a good δ exists is 2^{-m}. It doesn't take a very large m, to conclude with high confidence that $k = 0$.

C. Let $q(n)$ be the number of primitive gates in the n-qbit quantum Fourier transform gate. We have a recursive decomposition in which $S^{(n)}$ uses $n - 1$ swaps, $R^{(n)}$ uses $q(n - 1)$ gates, and $Q^{(n)}$ uses one H gate and $n - 1$ controlled one-qbit gates. Each controlled one-qbit gate uses seven primitive gates not including swaps. When the swaps are done efficiently, we need $2(n - 2)$ gates: $n - 2$ to bring the control qbit down the circuit where it's needed, and another $n - 2$ to bring it back up the circuit at the end. So the recurrence relation for $q(n)$ is

$$q(n) = (n - 1) + q(n - 1) + 1 + 7(n - 1) + 2(n - 2) = q(n - 1) + 10n - 11.$$

Because the difference $q(n) - q(n - 1)$ is linear, we conclude that $q(n)$ is $\mathcal{O}(n^2)$.

D. I see four requirements, although the first requirement sometimes goes without saying.

- $\tilde{Q}(\tilde{t})$ should be Hermitian for all $\tilde{t} \in [0, 1]$.
- The initial ground state $|\tilde{\omega}_0(0)\rangle$ should be preparable, meaning that we can put our quantum computer into that state easily.
- There should be a gap between the least two eigenvalues: $\tilde{\chi}_0(\tilde{t}) < \tilde{\chi}_1(\tilde{t})$ for all $\tilde{t} \in [0, 1]$.
- The final ground state $|\tilde{\omega}_0(1)\rangle$ should be the solution to our computational problem.

Under these conditions, we can start our computer in the ground state, and the adiabatic theorem says that it will stay in the ground state (approximately, and if run slowly enough), so at the end of our computation we will observe the solution to our problem with high probability.