# How much does it cost to build an iOS app?

- Costs
  - iOS Developer Rate
    - North America, Senior Level Developer is about $121-140.
      - Source: https://www.codementor.io/freelance-rates/ios-developers
    - Do I need to higher a "senior level" developer?
      - No, but with less experienced developers you take on more risk. If you are not technical yourself, that adds even more risk.
      - A developer will be making decisions constantly without getting input from the project owner or team. What to build, what not to build, how to do it all. You need to have trust they are making good decisions for the lifetime of this product.
    - How does a non-technical product owner evaluate potential developer hires?
      - Should I hire full time or contractors?
        - Benefits of Contractors
          - Pros
            - Can usually be hired for a short "discovery phase" with no long term commitment. This can be invaluable for a non-tech product owner looking to better learn the world they are about to enter -- and what it would be like to work with this individual.
            - Less financial commitment
            - Can be hired to solve specialized problems.
          - Cons
            - Many like to come in and rewrite the past, when this is usually a bad idea.
            - Similarly, many contractors don't have to live with their code for very long and could be lacking solid refactoring skills.
        - Benefits of Full Time employment
          - Pros
            - More hires will be in it for the long haul. They'll have to live with the code for a long time and this should make their decisions fall in line better with the long term stability of the project.
            - Can help dictate company culture.
          - Cons
            - Large financial commitment
              - Vacation Days
              - Sick Days
              - Health Insurance
              - Retirement Contributions
            - Takes longer to find the right people.
      - Things to look for
        - Has this person worked on long term successful projects?
          - What do the app reviews look like?
          - Download the app, play with them. Does it meet the quality standards you are looking for?
        - Can you see some samples of their work?
          - Do they do automated testing?
          - Have they invested in continuous integration?
          - Does their code have inline documentation?
        - Contact References
          - Most references provide will lean towards positive reactions but try to dig deeper. Ask hard questions about a disagreement that was had, times when hard truths were discussed.
        - Does this person communicate well?
          - Much of the job is not coding. Developers need to be able to communicate their ideas and concerns.
        - Research projects Listed on a their resume.
  - Other Roles
    - Product Designer
    - User Experience Designer
    - Backend Developer
      - Could be minimized if you can utilize things like Firebase but might be required depending on product needs.
    - Quality Assurance and Testers
    - Marketing Writers / Planners
      - Preparing App Store meta data, screenshots and videos.
      - Preparing public website meta data, screenshots and videos.
    - Developer Marketing Writers / Planners
      - If you are building a public SDK or something other developers themselves will consume.
    - Customer Support
    - Other Developers
      - If you can not provide an informed review of the developer's work, hire a second developer so code can be reviews and architectures evaluated. Letting a single developer make all the decisions adds high risk.
    - Legal Fees
      - Privacy Policy is required for all apps.
        - Source: https://developer.apple.com/app-store/review/guidelines/#privacy
    - Can a single person do all of this well?
      - Generally no. The smaller the team the more generalized the developer you are hiring needs to be though, so keep in mind there are lot of kinds of work to be done to get even a simple 1.0 app out the door.
  - Related Service Costs
    - Yearly Developer Accounts
    - Web Server Costs
    - Crash Report tools
    - App Store Revenue tools
    - Customer Feedback tools
- Development Loop
  - Speculate to identify a problem and solution.
  - Design a product that provides value toward solving this solution (in small parts).
  - Validate the design with user testing.
  - Build the solution.
  - Ship working code to users.
  - Validate the code with user testing.
  - Repeat this circle in the smallest possible circle. Trim scope without mercy.
- Other Notes
  - Be proud of what you build.
    - You'll probably have to live with this code for longer than you think. Don't skimp on quality / tested code. Instead cut scope and love the code you do produce.
    - Don't over develop. Solve the needs of today, and that need is customer validation. There is no point in building something the user will not use and pay for.
  - Don't Reinvent the Wheel
    - When using third-party solutions, try as best you can to abstract that specific solution. Don't let the tech dictate your code architecture. Make sure you can swap out this third party solution with another, within reason.
    - Example, if you need push notifications, don't build a push notification server, use a service that already does this.
    - Other big benefit of using an existing solution is that it will likely have way more documentation than is likely to be written about internally-made code. This will be helpful when you need to hire new or replacement developers.
  - App Rot
    - Code needs to be maintained as crash reports are collected, OS system updates are released, developer tools are updated. Even when you are satisfied with the state of the app, expect some quarterly updates if not more.
- TL;DR; -- Most quality apps cost between $100,000 to $1,000,000
  - Source: https://savvyapps.com/blog/how-much-does-app-cost-massive-review-pricing-budget-considerations
  - Big agency (6+ people) cost $10 million and more.
  - Smaller agency (3-6 people) cost between $100,000 to $2,500,000.
  - Smaller shop / freelancers (2-3 person team) $30,000 to $100,000
    - I would not recommend trying to hire a single developer for an app. At a minimum get a second developer for code review and to validate technical direction.
- Assumptions
  - Rate of software project failure is very high.
    - In order to protect ourselves from risk we must validate our solution as fast as possible.
    - We do not want to run out of money and have never shipped a product.
      - If you can afford 8 weeks of developer time, get something out the door in 2 weeks. Use the following 6 weeks for gradual improvement.
        - This is more work in total than if you were to compare it against the unrealistic idea that a designer can draw every screen of the perfect day-one untested solution and then the developer will build out each screen to arrive a v1.0, and thus ending the project / contract.
  - You could build the best user experience and technical foundation and the app will still fail to find an audience.
    - Poor or no marketing
    - Was not solving a problem people were looking to solve.
  - Instead of asking "How much to build a car?" the better question is what is the user pain and can I find a person or team that will help me discover the solution to this pain.
    - The pain is I need to get to and from the grocery store. We may not have to build a car. We might be able to build a bike (or even a skateboard for the right audience). OR a pivot and build a home deliver service. Keep asking why are we building this over and over.
- Common Wrong Assumptions
  - To arrive at the full project cost, I simply need to list features vs hourly estimates vs hourly costs.
    - Unless the technical scope of task is an afternoons worth of effort (and sometimes even then) this will never end well.
  - Once we ship a v1.0 I no longer need the contract developer.
  - We know what the user needs and will buy from us.
    - We don't.
  - We need to build iOS and Android at the same time.
    - Until you have successful designed a working solution this is probably a bad and wasteful decision. Pick a platform and solve problems, build value and then scale your solution by adding new platforms.
      - While I am personally biased towards native solutions, even if you want to utilize things like Flutter or React Native I would still pick iOS or Android and not both on day one.
      - One exception to this rule would be building a platform where users need to communicate with each other. If this is the case and you really want to do both then make a compromise and extremely scale back the scope to account for the extra development effort during this pre-"product-market fit" time.
      - This goes towards, tablets and phones too. Pick one size. Grow platform options later.